ΒΟΗΘΗΜΑ

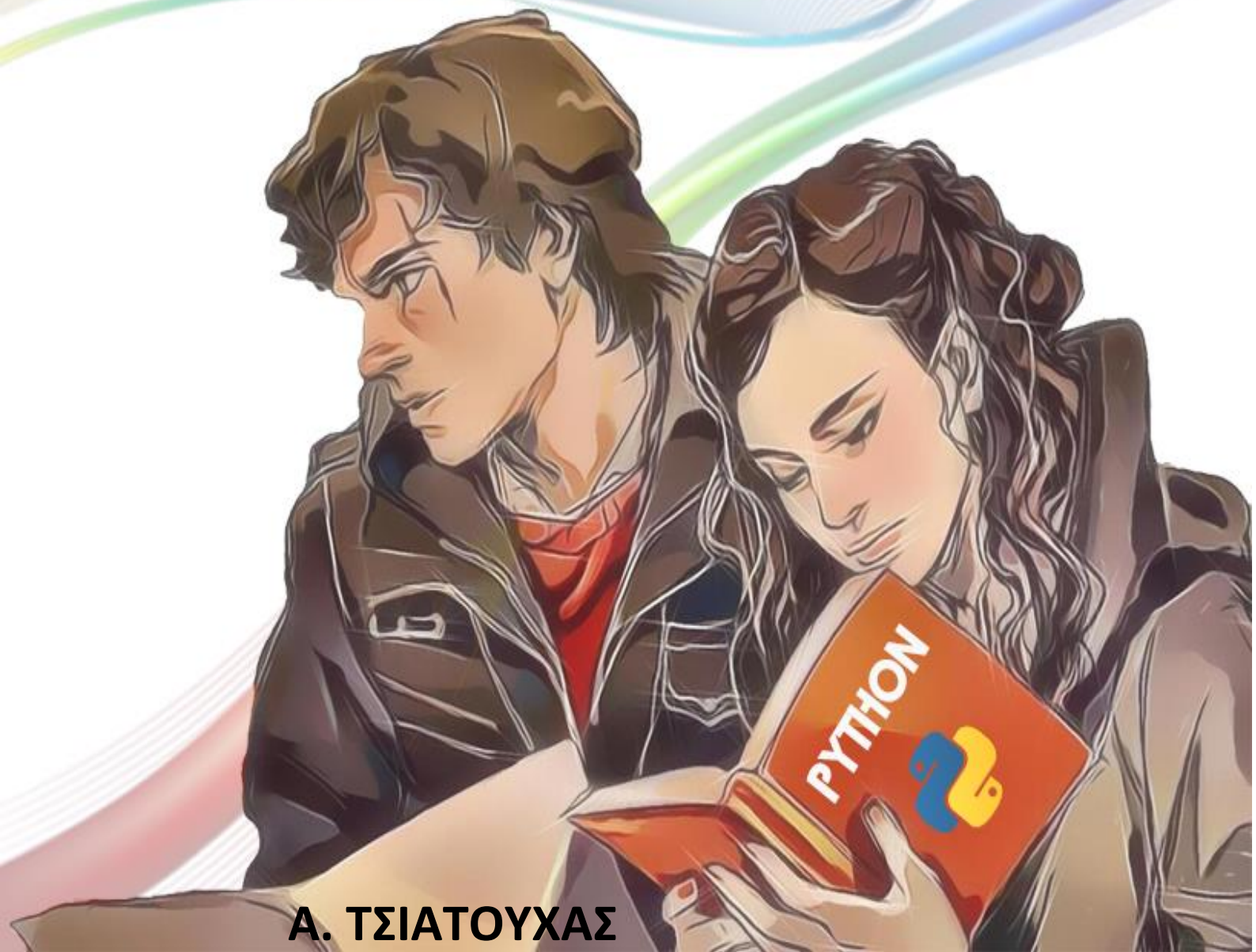
Γ' ΤΑΞΗ ΕΠΑ.Λ.

PYTHON

329

ΛΥΜΕΝΕΣ ΑΣΚΗΣΕΙΣ

ΕΠΙΠΕΔΟΥ ΠΑΝΕΛΛΑΔΙΚΩΝ ΕΞΕΤΑΣΕΩΝ

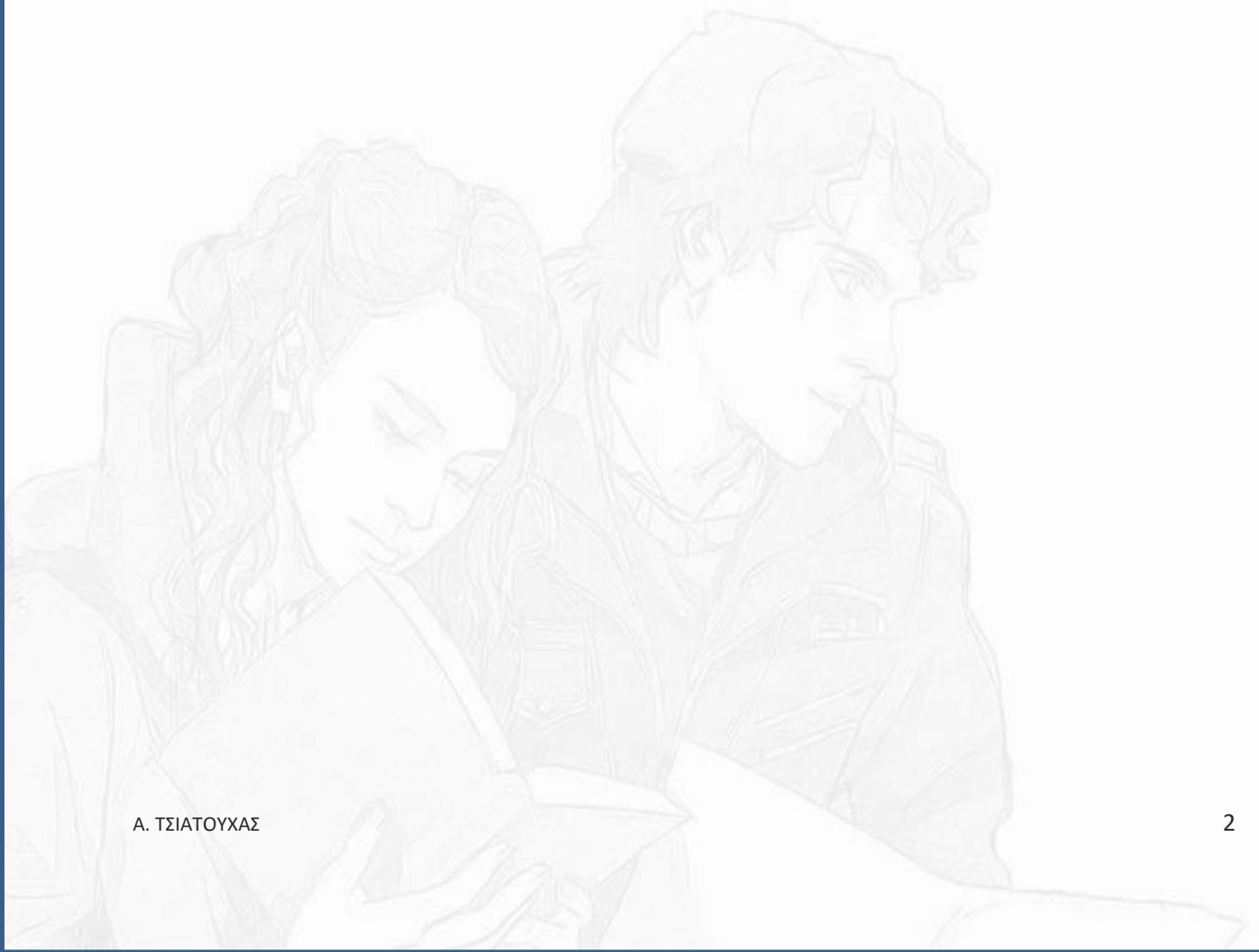


Α. ΤΣΙΑΤΟΥΧΑΣ

ΚΕΦΑΛΑΙΑ 3 & 4

ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

ΑΛΓΟΡΙΘΜΙΚΕΣ ΔΟΜΕΣ



ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑΙΟ 4

1) Να γραφτεί πρόγραμμα το οποίο θα διαβάζει την βαθμολογία των μαθητών ενός σχολείου μέχρι να διαβαστεί 0 (μηδέν) με έλεγχο εγκυρότητας τιμών (0 έως και 20). Στην συνέχεια να υπολογίζει και να εμφανίζει:

Α) Τον αριθμό των μαθητών

Β) Τον μεγαλύτερο βαθμό

Γ) Το ποσοστό των παιδιών που έμειναν στην ίδια τάξη με βαθμό μικρότερο ή ίσο από 9.5

2) Σε ένα κοντέινερ χωρητικότητας 300 κιλών φορτώνουμε παλέτες. Να γραφτεί πρόγραμμα που θα διαβάζει το βάρος των παλετών (σε κιλά) μέχρι να διαβάσει 0 ή αρνητικό αριθμό. Μετά να εμφανίζει κατάλληλο μήνυμα ανάλογα με το αν το βάρος των παλετών χωράει στο κοντέινερ.

Στην περίπτωση που χωράνε στο κοντέινερ να τυπωθεί το βάρος της παλέτας που είναι η βαρύτερη.

3) Στο υπεραστικό ΚΤΕΛ υπάρχουν 4 κατηγορίες εισιτηρίων:

Πολύτεκνων (Π), Αναπηρικό (Α), Φοιτητικό (Φ) και κανονικό (Κ).

Οι δύο πρώτες κατηγορίες πληρώνουν 40% της αξίας του κανονικού εισιτηρίου.

Η τρίτη κατηγορία έχει έκπτωση 15%

Η τέταρτη κατηγορία πληρώνει ολόκληρη την αξία που είναι 20€.

Να γραφτεί πρόγραμμα που για ένα λεωφορείο 50 θέσεων θα διαβάζει την κατηγορία κάθε επιβάτη και :

Α) θα εμφανίζει και κάθε επιβάτη τα χρήματα που πρέπει να πληρώσει

Β) θα εμφανίζει τα συνολικά χρήματα που έχουν εισπραχθεί

Γ) Το πλήθος των θέσεων που καλύφθηκαν

Το πρόγραμμα θα τερματίζει όταν δοθεί ως κατηγορία το γράμμα Χ ή όταν το λεωφορείο γεμίσει.

ΑΣΚΗΣΗ 1

```
V=input('Dose Vathmo')
while V<0 or V>20:
     V=input('Dose SOSTO Vathmo')
c=0
c2=0
max1=0
while  V!=0:
   c=c+1

   if V>max1:
      max1=V

   if V<=9.5:
      c2=c2+1

   V=input('Dose Vathmo')
   while V<0 or V>20:
         V=input('Dose SOSTO Vathmo')

p= 100.0*c2/c

print   "Aritmos mathiton=", c
print   "Megaliteri vathmologia=", max1
print   "pososto apotixias=", p, " %"
```

ΑΣΚΗΣΗ 2

```
C=500
S=0
max1=0

B=input('Dose Varos')

while B>0:

   S=S+B

   if B>max1:
      max1=B

   B=input('Dose Varos')

if S>C:
   print "Ypervaro"
else:
   print max1
```

ΑΣΚΗΣΗ 3

```
C=20
E=0
S=0

K=raw_input("Dose katigoria")

while   K!="X"   and    C>0:

   C=C-1

   E=E+1

   if   K=="P"   or   K=="A":
      Xr=20*40/100

   if K=="F":
      Ekprosi=20*15/100
      Xr= 20-Ekptosi

   if K=="K":
      Xr=20

   S=S+Xr
   print Xr

print   "Synolika xrimata=",  S
print   "Theseis pou kalifthikan=",  E
```

ΑΣΚΗΣΗ 1 - Δραστηριότητα 29 βιβλίου.

Ένα ασανσέρ έχει μέγιστο όριο ασφάλειας τα 500 κιλά. Να γράψετε πρόγραμμα σε Python που θα διαβάζει το βάρος και τη σειρά με την οποία κάθε άτομο εισέρχεται στο ασανσέρ (π.χ. 45 1, 89 2). Το πρόγραμμα θα τερματίζει, όταν το ασανσέρ γεμίσει (σε σχέση με το μέγιστο επιτρεπτό όριο ασφαλείας). Στη συνέχεια, θα εμφανίζει τη σειρά του τελευταίου ατόμου που κατάφερε να μπει στο ασανσέρ.

ΑΣΚΗΣΗ 2 - Δραστηριότητα 30 βιβλίου.

Σε ένα πλοίο υπάρχουν εισιτήρια Α' Θέσης (κωδικός 0) προς 50€ και Β' θέσης (κωδικός 1) προς 20€, το ένα. Ο μέγιστος επιτρεπόμενος αριθμός επιβατών είναι 400 άτομα και θεωρούμε ότι τελικά το πλοίο γέμισε για το συγκεκριμένο προορισμό που εξετάζουμε.

Να γράψετε πρόγραμμα σε Python, το οποίο να:

1) διαβάζει την κατηγορία εισιτηρίου (κωδικός 0 ή 1) για κάθε επιβάτη

2) εμφανίζει το πλήθος των επιβατών της Α' θέσης

3) εμφανίζει το συνολικό ποσό που πλήρωσαν όλοι οι επιβάτες.

ΑΣΚΗΣΗ 3 - Δραστηριότητα 33 βιβλίου.

Να γράψετε, σε Python, πρόγραμμα με το οποίο:

Να καταχωρούνται επαναληπτικά: ο αριθμός κυκλοφορίας οχήματος και ποσό κλήσης από παρκάρισμα ή άλλη αιτία, με την καταχώρηση να επαναλαμβάνεται μέχρι να δοθεί αριθμός κυκλοφορίας 99.

Στο τέλος να εμφανίζει:

1. Το πλήθος των οχημάτων που καταχωρήθηκαν.

2. Το συνολικό ποσό κλήσεων που θα εισπραχθεί.

3. Τον αριθμό κυκλοφορίας του τελευταίου από τα οχήματα που έλαβαν το μέγιστο ποσό κλήσης, καθώς και το ποσό της κλήσης αυτής.

ΑΣΚΗΣΗ 1

```
W = input(" Βάρος ατόμου σε κιλά = ")
pos = input(" Σειρά = ")
persons = 0
C = 500
while     W <= C :
        C = C – W
       persons = persons + 1
       pos = input(" Σειρά = ")
      W = input(" Βάρος ατόμου σε κιλά = ")
print   " Σειρά τελευταίου ", pos
```

ΑΣΚΗΣΗ 2

```
E = 1
while      E <= 400 :
          cat = input(" Κατηγορία εισιτηρίου (0/1) = ")
          E = E + 1
          if      cat == 0 :
                  k = k + 1
print " Επιβάτες Α’ θέσης : ", k
total = k * 50 + (400 - k)*20
print " Συνολικό ποσό " , total
```

ΑΣΚΗΣΗ 3

```
N = input(" Επόμενος αριθμός οχήματος = ")
V = 0
total = 0
max = 0
max_car = N
while      N != 99 :
           K = input(" Ποσό κλήσης= ")
           total = total + K
           if    K > max :
                 max = K
                 max_car = N
           elif    K == max :
                 max_car = N
           V = V +1
           N = input(" Επόμενος αριθμός οχήματος = ")

print " Πλήθος οχημάτων : ", V
print " Συνολικό Ποσό : ", total
print " Μέγιστο ποσό : ", max
print " τελευταίο όχημα με μέγιστο ποσό : ", max_car
```

1) Ένα τυπογραφείο χρεώνει κλιμακωτά τους πελάτες του ως εξής:

· Για το πρώτα 100 βιβλία [1-100] προς 2.80 λίρες το ένα.

· Για τα επόμενα 400 βιβλία [101-500] προς 2.40 λίρες το ένα.

· Για τα υπόλοιπα βιβλία (περισσότερα από 500) προς 1.50 λίρες το ένα.

Να γραφεί πρόγραμμα σε Python που θα υπολογίζει συνολικά τι πρέπει να πληρώσει κάποιος που θέλει να εκτυπώσει κάποια βιβλία.

2) Να γράψετε πρόγραμμα όπου θα δίνετε τον αριθμό των αυτοκινήτων που περνούν από ένα φανάρι (Ν).

Έστω ότι υπάρχουν αυτοκίνητα με τρία μόνο χρώματα (Άσπρα, Μαύρα και Κόκκινα).

Να δίνετε για κάθε αυτοκίνητο το χρώμα του (το γράμμα Α για τα άσπρα, το Μ για τα μαύρα και το Κ για τα κόκκινα).

Το πρόγραμμα να υπολογίζει και να τυπώνει τον συνολικό αριθμό των Άσπρων αυτοκινήτων.

Το πρόγραμμα να υπολογίζει και να τυπώνει τον συνολικό αριθμό των Μαύρων αυτοκινήτων.

Το πρόγραμμα να υπολογίζει και να τυπώνει τον συνολικό αριθμό των Κόκκινων αυτοκινήτων.

Να τυπώνει το χρώμα με τα περισσότερα αυτοκίνητα

Να χρησιμοποιηθεί η δομή While.

3) Να δίνετε από το πληκτρολόγιο τον βαθμό στο μάθημα του προγραμματισμού. Να γίνεται έλεγχος ορθότητας τιμών ώστε ο βαθμός να είναι μεταξύ 0 και 20.

4) Να δίνετε από το πληκτρολόγιο τον αριθμό των παιχτών μπάσκετ που παίρνουν μέρος στο τουρνουά μπάσκετ Ακρόπολης.

Για κάθε παίχτη να δίνετε το ύψος του (πραγματικός).

Το πρόγραμμα να υπολογίζει και να τυπώνει τον αριθμό παιχτών με ύψος μεγαλύτερο από 2m.

Να γίνεται έλεγχος ορθότητας τιμών ώστε το ύψος των παιχτών να είναι μεταξύ 1.80 και 2.20

ΑΣΚΗΣΗ 2

```
N=input('Dose arithmo car')
c1=0
c2=0
c3=0
x=1
while x <=N:
   C=raw_input('Dose xroma')
   while C not in ["A", "M", "K"] :
      C=raw_input('Dose SOSTO xroma')

   if C=="A":
      c1=c1+1
   if C=="M":
      c2=c2+1
   if C=="K":
      c3=c3+1
   x=x+1
print "Aspra: ", c1
print "Mavra: ", c2
print "Kokkina: ", c3

max1=c1
Cr="Aspro"
if c2>max1:
   max1=c2
   Cr="Mavro"
if c3>max1:
   max1=c3
   Cr="Kokkino"

print "perissotera ", Cr
```

ΑΣΚΗΣΗ 1

```
B=input('Dose arithmo vivlion')
if B>=1 and B<=100:
   Xr=B*2.80
if B>=101 and B<=500:
   Xr=100*2.80 + (B-100)*2.40
if B>500:
   Xr=100*2.80 + 400*2.40 + (B-500)*1.50
print   Xr
```

ΑΣΚΗΣΗ 3

```
V=input('Dose Vathmo')
while V<0 or V>20:
     V=input('Dose SOSTO Vathmo')
```

ΑΣΚΗΣΗ 4

```
N=input('Dose arithmo pexton')
c=0
for x  in range (N):
   H=float(input('Dose ypsos'))
   while H<1.80 or H>2.20 :
      H=float(input('Dose SOSTO ypsos'))
   if H>2:
      c=c+1

print c
```

1) Να γραφεί πρόγραμμα που να διαβάζει ένα άγνωστο πλήθος ακεραίων θετικών αριθμών. Αν διαβαστεί αρνητικός ή μηδέν να ενημερώνεται ο χρήστης με σχετικό μήνυμα και να προτρέπεται να ξαναδώσει αριθμό. Στην συνέχεια όταν δοθεί ο αριθμός 10 να σταματάει και να εμφανίζει τον μεγαλύτερο από αυτούς που διάβασε.

2) Σε ένα αγώνα ρίψης ακοντίου , διεξάγεται ο προκριματικός γύρος με τη συμμετοχή **14 αθλητών**. Στην τελική φάση προκρίνονται όσοι αθλητές επιτύχουν επίδοση άνω των 80 μέτρων. Να γραφεί αλγόριθμος ο οποίος :

α) να διαβάζει το όνομα και την επίδοση κάθε αθλητή, να υπολογίζει και να εμφανίζει τα ονόματα και το πλήθος των αθλητών που πέρασαν το όριο.

β) να εμφανίζει το όνομα του αθλητή που πλησίασε πιο κοντά από όλους τα 95 μέτρα.

3) Στο άθλημα της Άρσης Βαρών οι αθλητές συμμετέχουν στο «Αρασέ». Κάθε αθλητής, δικαιούται να εκτελέσει **τρεις** προσπάθειες. Αφού ολοκληρώσει και τις τρεις προσπάθειες, τότε καταμετράται η προσπάθεια με το μεγαλύτερο βάρος για την κίνηση αυτή. Νικητής ανακηρύσσεται εκείνος που θα πετύχει το μεγαλύτερο βάρος. Σε περίπτωση ίδιας επίδοσης μεταξύ αθλητών τη πρώτη θέση κερδίζει ο ελαφρύτερος σε βάρος.

Να γίνει αλγόριθμος που για 18 αθλητές, θα διαβάζει :

α) το όνομά τους, β) το βάρος τους, και γ) τα βάρη που σήκωσαν σε κάθε μία από τις προσπάθειές τους και θα εμφανίζει το όνομα του νικητή.

ΑΣΚΗΣΗ 1

```
max1=0

x=int(input('dose arithmo'))
while x<=0:
   print "Edoses lathos ariumo"
   x=int(input('dose sosto arithmo'))

while x!=10:
   if x>max1:
      max1=x

   x=int(input('dose arithmo'))
   while x<=0:
      print "Edoses lathos ariumo"
      x=int(input('dose sosto arithmo'))

print max1
```

ΑΣΚΗΣΗ 2

```
c=0
min1=95
Name=""

for x  in range (14):
   On=raw_input('Dose onoma')
   R=input("Dose epidosi athliti")

   if R>80:
      print On
      c=c+1

   if 95-R<min1:
      min1=95-R
      Name=On

print "Perasan =", c , "athlites"
print "Koditera sta 95 =", Name
```

ΑΣΚΗΣΗ 3

```
max2=0

for x  in range (18):
   On=raw_input('Dose onoma')
   V=input("Dose Varos athliti")
   B1=input("Dose 1h prospatheia")
   B2=input("Dose 2h prospatheia")
   B3=input("Dose 3h prospatheia")

   max1=B1
   if B2>max1:
      max1=B2
   if B3>max1:
      max1=B3

   if max1>max2:
      max2=max1
      name=On
      Varos=V

   if max1==max2:
      if V<Varos:
         name=On
         Varos=V

print "O nikitis einai=",name
```

1) Ένας αγρότης ανάλογα με τα στρέμματα που διαθέτει πληρώνει φόρο κλιμακωτά σύμφωνα με τον παρακάτω πίνακα:

Στρέμματα	**Φόρος ανά στέμμα**
1-100	1€
Πάνω από 100 – 300	2€
Πάνω από 300	3€

Σε ένα χωριό υπάρχουν 100 αγρότες.

Να γράψετε πρόγραμμα σε python το οποίο για κάθε ένα αγρότη:

α) Να διαβάζει το ονοματεπώνυμό του.

β) Στη συνέχεια να διαβάζει τον κωδικό αριθμό (ένας ακέραιος αριθμός) του κάθε χωραφιού και τα στρέμματα μέχρι να μας δοθεί για κωδικός ο αριθμός 0, και να υπολογίζει και να εμφανίζει τα συνολικά στέμματα που έχει ο συγκεκριμένος αγρότης.

γ) Να υπολογίζει και να εμφανίζει το συνολικό φόρο που πρέπει να πληρώσει ο συγκεκριμένος αγρότης.

δ) Να υπολογίζει και να εμφανίζει πόσο φόρο θα πληρώσουν όλοι μαζί οι αγρότες του χωριού.

ε) Να υπολογίζει και να εμφανίζει κατά μέσο όρο πόσο φόρο θα πληρώσει ο κάθε αγρότης.

στ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το μεγαλύτερο φόρο και το φόρο αυτό.

ζ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το μικρότερο φόρο και το φόρο αυτό.

ΑΣΚΗΣΗ 1

```
SUMF=0
MAXF=0
MAXFONOMA=""
MINF=1000000000
MINFONOMA=""

for i in range(100):
    ONEP=raw_input("Δώσε το ονοματεπώνυμο")
    SUMA=0
    k=int(input("Δώσε τον κωδικό αριθμό"))
    while k!=0:
        s=float(input("Δώσε τα στρέμματα"))
        SUMA=SUMA+s
        k=int(input("Δώσε τον κωδικό αριθμό"))
     print "Ο αγρότης ",ONEP," έχει ",SUMA," στρέμματα"

     if SUMA>=1 and SUMA<=100:
          f=SUMA*1
     elif  SUMA>100 and SUMA<=300:
         f=100*1+(SUMA-100)*2
     elif SUMA>300:
         f=100*1+200*2+(SUMA-300)*3
     print "Ο αγρότης ",ONEP," πρέπει να πληρώσει φόρο ",f," €"

     SUMF=SUMF+f

     if   f>MAXF:
         MAXF=f
         MAXFONOMA=ONEP

     if   f<MINF:
         MINF=f
         MINFONOMA=ONEP
print "Όλοι οι αγρότες του χωριού θα πληρώσουν ",SUMF," φόρο"

MO=SUMF/100.0
print "Κάθε αγρότης θα πληρώσει κατά μέσο όρο ",MO," €"

print "Ο αγρότης ",MAXFONOMA," έχει το μεγαλύτερο φόρο που είναι ",MAXF

print "Ο αγρότης ",MINFONOMA," έχει το μικρότερο φόρο που είναι ",MINF
```

1) Γίνεται μία έρευνα για το πόσα λεπτά χρησιμοποιούν καθημερινά οι μαθητές τον ηλεκτρονικό υπολογιστή για να παίξουν παιχνίδια. Για το σκοπό αυτό ρωτήθηκαν μαθητές με ηλικία από 10 μέχρι και 18 ετών.

Να γράψετε πρόγραμμα σε python το οποίο :

α) Να διαβάζει για κάθε μαθητή/τρια το φύλο (Α για αγόρι και Κ για κορίτσι), την ηλικία του και τα λεπτά που παίζει συνήθως σε μία ημέρα

β) Η διαδικασία να επαναλαμβάνεται μέχρι να μας δοθεί για φύλο η λέξη “TELOS”

γ) Να γίνει έλεγχος ορθότητας για την ηλικία, ότι δηλαδή η ηλικία των μαθητών που καταχωρίζεται να είναι από 10 έως και 18.

δ) Να εμφανίζει το κατάλληλο μήνυμα σύμφωνα με τον παρακάτω πίνακα:

Λεπτά	**Εξάρτηση από ηλεκτρονικά παιχνίδια**
Κάτω από 15	Μικρή
από15 μέχρι και 30	Μέτρια
Πάνω από 30 μέχρι και 60	Μεγάλη
Πάνω από 60	Πάρα πολύ μεγάλη

ε) Να υπολογίζει και να εμφανίζει πόσοι/ες μαθητές/τριες έχουμε από κάθε κατηγορία

στ) Να υπολογίζει και να εμφανίζει το σύνολο των λεπτών που παίζουν όλοι οι μαθητές

ζ) Να υπολογίζει και να εμφανίζει το Μέσο όρο όλων των μαθητών

η) Να υπολογίζει και να εμφανίζει τα περισσότερα λεπτά που έπαιξε κάποιος/α

θ) Να υπολογίζει και να εμφανίζει τα λιγότερα λεπτά που έπαιξε κάποιος/α

ι) Να υπολογίζει και να εμφανίζει το σύνολο των λεπτών που έπαιξαν τα αγόρια

ια) Να υπολογίζει και να εμφανίζει το ΜΟ των λεπτών που έπαιξαν τα αγόρια

ιβ) Να υπολογίζει και να εμφανίζει το σύνολο των λεπτών που έπαιξαν τα κορίτσια

ιγ) Να υπολογίζει και να εμφανίζει το ΜΟ των λεπτών που έπαιξαν τα κορίτσια

ιδ) (Δύσκολο) Να υπολογίζει και να εμφανίζει το ποσοστό των παιδιών που παίζουν πάνω από 60 λεπτά την ημέρα στον ηλεκτρονικό υπολογιστή.

ΑΣΚΗΣΗ 1

```
m1=m2=m3=m4=0
SUM=0.0
MAX=0
MIN=1500
SUMA=0.0
mA=0
SUMK=0.0
mK=0
m60=0

f=raw_input("Δώσε το φύλο")
while f!="TELOS":

   h=input("Δώσε την ηλικία σε έτη")

   while h<10 or h>18:
      h=input("Δώσε την ηλικία σε έτη")

   lep=float(input("Δώσε τα λεπτά που παιζει συνήθως σε μία ημέρα"))

   if lep<15:
      print "Εξάρτηση Μικρή"
      m1=m1+1
   if lep>=15 and lep<=30:
      print "Εξάρτηση Μέτρια"
      m2=m2+1
   if lep>30 and lep<=60:
      print "Εξάρτηση Μεγάλη"
      m3=m3+1
   if lep>60 :
      print "Εξάρτηση Πάρα πολύ μεγάλη"
      m4=m4+1

   SUM=SUM+lep

   if lep>MAX:
      MAX=lep

   if lep<MIN:
      MIN=lep


   if f=="A":
      SUMA=SUMA+lep
      mA=mA+1
   else:
```

```
      SUMK=SUMK+lep
      mK=mK+1

   if lep>60:
      m60=m60+1

   f=raw_input("Δώσε το φύλο")

print " Οι μαθητές παίζουν συνολικά κάθε μέρα ", SUM ," λεπτά"

pl=m1+m2+m3+m4

if pl==0:
   print "Δεν υπάρχει Μέσος όρος γιατί δεν μου έδωσες κανένα μαθητή/τρια"
else:
   MO=SUM/pl
   print "Τα παιδιά παίζουν κατά μέσο όρο ", MO ," λεπτά"

print " Οι μαθητές/τριες που έχουν μικρή εξάρτηση είναι: ", m1
print " Οι μαθητές/τριες που έχουν μέτρια εξάρτηση είναι: ", m2
print " Οι μαθητές/τριες που έχουν μεγάλη εξάρτηση είναι: ", m3
print " Οι μαθητές/τριες που έχουν Πάρα πολύ μεγάλη εξάρτηση είναι: ", m4

print "Τα περισσότερα λεπτά που έπαιξε κάποιος/α μαθητής/τρια είναι",MAX," λεπτά"
print "Τα λιγότερα λεπτά που έπαιξε κάποιος/α μαθητής/τρια είναι",MIN," λεπτά"

if mA==0:
   print "Δεν υπάρχει MO αφού δεν έδωσες αγόρια"
else:
   MOA=SUMA/mA
   print "Ο Μέσος όρος των λεπτών που παίζουν τα αγόρια είναι ", MOA

X=(float(m60)/(mA+mK))*100
print "Το ποσοστό των παιδιών που παίζουν πάνω από 60 λεπτά την ημέρα είναι",X," %"
```

ΚΕΦΑΛΑΙΟ 5 & 8

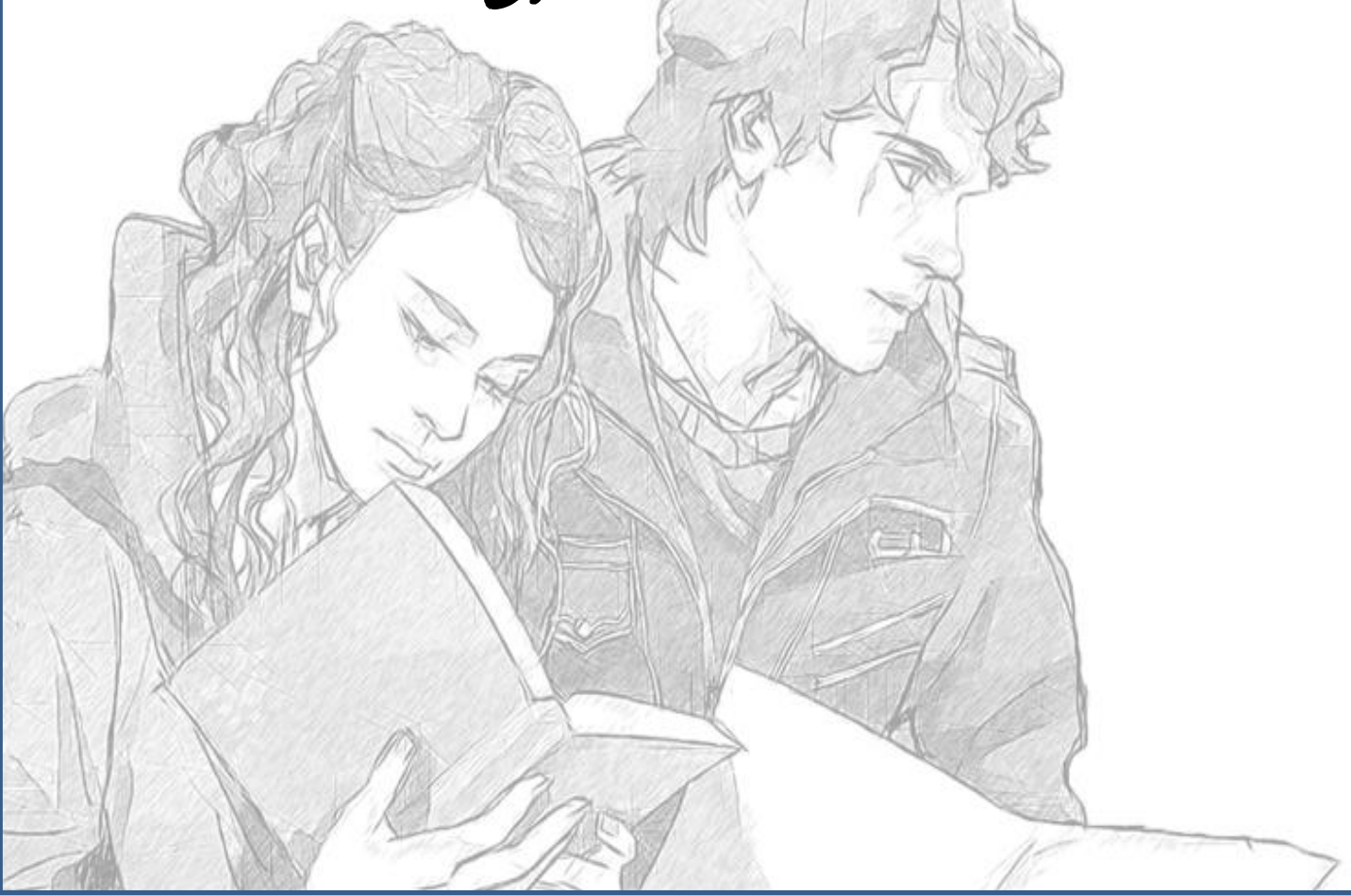
ΚΛΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΙΙ

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΙΙ

ΛΙΣΤΕΣ

&

ΣΥΜΒΟΛΟΣΕΙΡΕΣ



ΑΣΚΗΣΕΙΣ ΛΙΣΤΕΣ - ΣΥΜΒΟΛΟΣΕΙΡΕΣ

1) Να δηλώσετε την μεταβλητή word ίση με την λέξη «olokaftoma».

Να τυπώσετε το 1ο το 3ο και το 8ο γράμμα του word

Να τυπώσετε τον αριθμό των γραμμάτων του word

Αν ο αριθμός των γραμμάτων του word είναι μεγαλύτερος από 8 τότε να τυπωθεί «kanei gia kodikos».

Να τυπώσετε το εξής (την λέξη από το τέλος προς την αρχή):

a
m
o
t
f
a
k
o
l
o

Να τυπώσετε τα πέντε πρώτα γράμματα του word

Να τυπώσετε τα πέντε τελευταία γράμματα του word

Να τυπωθεί η λέξη olokaftoma με κεφαλαία γράμματα

Αν το word περιέχει το γράμμα “t” να τυπωθεί «Έχει t» αλλιώς να τυπωθεί «Δεν έχει t»

Να τυπωθεί η θέση του γράμματος “f” στο word

Πόσα “o” περιέχονται στο word

ΑΣΚΗΣΗ 1

```
word="olokaftoma"

print word[0], word[2], word[7]

print len(word)

if len(word)>8:
   print 'kanei gia kodikos'

for x in range(len(word)-1, -1,-1):
   print word[x]

print word[ : 5]

print word[5 : ]

print word.upper()

if "t" in word:
   print "exei t"

else:
   print "den exei t"

print word.find("f")

print word.count("o")
```

1) Να δίνετε μια συμβολοσειρά από το πληκτρολόγιο.

Α) Να τυπωθεί ανεστραμμένη η λέξη (με δύο τρόπους)

Β) Να υπολογιστούν και να τυπωθούν πόσα σύμφωνα υπάρχουν στην συμβολοσειρά (Η συμβολοσειρά περιέχει φωνήεντα και σύμφωνα μόνο)

2) Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει μία λέξη και θα εμφανίζει πόσα κεφαλαία αγγλικά γράμματα περιέχει η λέξη.

3) Να γράψετε πρόγραμμα στη γλώσσα Python το οποίο θα δέχεται ως είσοδο ένα κείμενο και θα εμφανίζει πόσες φορές εμφανίζεται κάθε γράμμα του αγγλικού αλφαβήτου σε αυτό.

4) Να δίνετε μια συμβολοσειρά από το πληκτρολόγιο.

Α) Να τυπωθεί ο αριθμός των προτάσεων του κειμένου. (Κάθε πρόταση τελειώνει με τελεία (.))

Β) Να υπολογιστεί και να τυπωθεί πόσες φορές υπάρχει το όνομα «Maria» μέσα στο κείμενο. Δεν μπορεί να βρίσκεται στην αρχή και στο τέλος μιας πρότασης.

5) Να γράψετε μια συνάρτηση, η οποία θα δέχεται μια λίστα από λέξεις και θα επιστρέφει τη λέξη με τα περισσότερα φωνήεντα.

6) Να αναπτύξετε πρόγραμμα που αφού διαβάσει μια συμβολοσειρά με τουλάχιστον 40 χαρακτήρες

Α. να τυπώνει πόσες φορές εμφανίζεται ο χαρακτήρας «a» σε περιττές θέσεις της συμβολοσειράς.
Β. να τυπώνει πόσες φορές εμφανίζεται ο χαρακτήρας «a» σε άρτιες θέσεις της συμβολοσειράς.

ΑΣΚΗΣΗ 1

```
A=raw_input("Dose leksi")
B=""
for x in A:                                  # 1 τρόπος
   B=x+B
print B

for x in range(len(A)-1, -1,-1):             # 2 τρόπος
   print A[x],

print ""
F="AEIOYHaeioy"
c=0
for x in A:
   if x not in F:
      c=c+1
print c
```

ΑΣΚΗΣΗ 2

```
A=raw_input('Dose leksi')

c=0

for x  in A:
   if x  in A.upper():
      c=c+1

print c
```

ΑΣΚΗΣΗ 3

```
A=raw_input('Dose leksi')

B="abcdefghijklmnopqrstuvwxyz"


for i  in range(len(B)):
   c=0
   for x  in A:
      if x  == B[i]:
         c=c+1
   print " To ", B[i], " vrethike ", c, "fores"
```

ΑΣΚΗΣΗ 4

```
A=raw_input("Dose keimeno")

c=0
for x  in A:
   if x==".":
      c=c+1

print "Arithmos protaseon = ", c

c1=0
for x in range(0, len(A)-6,1):
   print A[x : x+7]
   if A[x : x+7] == " Maria ":
      c1=c1+1

print  c1
```

ΑΣΚΗΣΗ 5

```
A=["Podikoulis", "Parthenonas", "Asterismos", "Potamoploio", "Tromeros"]
F="AEIOYHaeioyh"
max1=0
for X in A:
   c=0
   for i  in X:
      if i  in F:
         c=c+1
   if c > max1:
      max1=c
      leksi=X

print X
```

ΑΣΚΗΣΗ 6

```
A=raw_input('Dose simvoloseira')
while len(A)<40:
   A=raw_input('Dose simvoloseira')

c1=0
for x  in range(1,len(A),2):
   if A[x]=="a":
      c1=c1+1
print c1
c2=0
for x  in range(0,len(A),2):
   if A[x]=="a":
      c2=c2+1
print c2
```

1)

1α. Σας δίνετε η συμβολοσειρά : A=”NEOS KOSMOS”

1β. Να τυπώσετε κατακόρυφα την συμβολοσειρά

1γ. Να τυπώσετε το μέγεθος της συμβολοσειράς (τον αριθμό των χαρακτήρων της)

1δ. Να τυπώσετε το πρώτο και το τελευταίο γράμμα της συμβολοσειράς

1ε. Να τυπώσετε την λέξη “KOSMOS”

1ζ. Χωρίς να χρησιμοποιήσετε κάποια μέθοδο ή συνάρτηση να γράψετε δικό σας κώδικα ώστε:

Α) Να τυπώσετε τον αριθμό των S της συμβολοσειράς

Β) Να τυπώσετε την συμβολοσειρά έχοντας αφαιρέσει όλα τα S και τα M

Γ) Να τυπώσετε πόσα φωνήεντα έχει η συμβολοσειρά

Δ) Να διαβάσετε την συμβολοσειρά και να την αποθηκεύσετε σε μια νέα συμβολοσειρά αντιστρέφοντας την σειρά των γραμμάτων δηλαδή από το τέλος προς την αρχή

π.χ. “SOMSOK SOEN”

1η. Σας δίνετε μια συμβολοσειρά : A= “Laser Toner Cartridge”

Να αποθηκεύσετε σε ένα ΑΡΧΕΙΟ με όνομα «Laser.txt», σε ξεχωριστές γραμμές, τις τρείς λέξεις της συμβολοσειράς Α. Στο τέλος του αρχείου να γραφτεί ο αριθμός των χαρακτήρων της συμβολοσειράς Α.

ΑΣΚΗΣΗ 1

```
A="NEOS KOSMOS"

for x in A:
   print x

print len(A)
print A[0], A[-1]

print A[5 :]

c=0
for x in A:
   if x=="S":
      c=c+1
print "Συνολο S", c

B=""
for x in A:
   if x!="S" and x!="M":
      B=B+x
print B

F="AEOIYHaeoiyh"
c=0
for x in A:
   if x in F:
      c=c+1
print "Συνολο φωνηέντων", c

D=""
for x in A:
   D= x + D
print D


A="Laser Toner Cartridge"

f1=open("Laser.txt", "w")
M=""
for x in A:
   if x !=" ":
      M=M+x

   if x==" ":
      f1.write(M + "\n")
      M=""

f1.write(M + "\n")

f1.close()
```

1) Να γράψετε μια συνάρτηση η οποία θα ελέγχει αν μια συμβολοσειρά αποτελεί ηλεκτρονική διεύθυνση αλληλογραφίας ελληνικού ιστότοπου, δηλαδή περιέχει το σύμβολο'@', δεν περιέχει κενά και έχει κατάληξη'.gr'.

2) Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει λέξεις από το πληκτρολόγιο, θα μετράει και θα εμφανίζει πόσες λέξεις ξεκινούν από το γράμμα Α (κεφαλαίο ή μικρό). Όταν δοθεί λέξη που να τελειώνει σε «Ω» ή «ω», τότε το πρόγραμμα θα τερματίζει.

3) Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει λέξεις από το πληκτρολόγιο και θα τις ενώνει σε μια μεγάλη πρόταση την οποία στη συνέχεια, θα εμφανίζει στην οθόνη. Οι λέξεις θα χωρίζονται μεταξύ τους με κενά και η πρόταση θα τελειώνει με τελεία '.' .

Σημείωση: Η εισαγωγή των λέξεων τελειώνει όταν δοθεί ο χαρακτήρας τελεία ".".

4) Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει μια λέξη και θα ελέγχει αν είναι παλινδρομική (καρκινική), δηλαδή αν διαβάζεται το ίδιο και αντίστροφα, για παράδειγμα radar, madam.

5) Σας δίνετε η λίστα Α=["Athens", "Paris", "London"]. Να αντικατασταθεί το πρώτο γράμμα κάθε πόλης της λίστας Α με το γράμμα "W".

6) Να δίνετε από το πληκτρολόγιο τα ονόματα 25 μαθητών και να τα τοποθετείται σε μια λίστα Α.

Α) Να τυπώσετε το μεγαλύτερο σε αριθμό χαρακτήρων όνομα.

Β) Να ελέγξετε αν υπάρχει συνωνυμία και να τυπώσετε αυτό το όνομα

Γ) Να τυπώσετε το όνομα με τα περισσότερα 'Α' και 'α'.

Δ) Να τυπώσετε τα ονόματα που έχουν κατάληξη «akis» και πιθανόν να έχουν καταγωγή από την Κρήτη.

7) Να πληκτρολογείτε μια συμβολοσειρά και στην συνέχεια να τοποθετείτε τους χαρακτήρες της σε αντίστροφη σειρά μέσα σε μια λίστα π.χ. "JAVA" → ["A", "V", "A", "J"]

8) Να δίνετε από το πληκτρολόγιο μια συμβολοσειρά και να τυπώνεται πόσα **σύμφωνα** έχει η συμβολοσειρά (Δεν υπάρχουν αριθμοί, κενά και ειδικοί χαρακτήρες)

9) Δημιουργήστε μια συνάρτηση find_New() (να δίνετε για είσοδο μια συμβολοσειρά και έναν χαρακτήρα και να επιστρέφει την θέση του χαρακτήρα στην συμβολοσειρά)

10) Δημιουργήστε μια συνάρτηση count_New() (να δίνετε για είσοδο μια συμβολοσειρά και έναν χαρακτήρα και να επιστρέφει τον αριθμό των χαρακτήρων στην συμβολοσειρά)

ΑΣΚΗΣΗ 1

```
def   email(EM):
        if "@" in EM:
                c=0
                for  x in EM:
                        if x=="@":
                                c=c+1
        if   c!=1   or " " in EM   or   EM[len(EM)-3 : ]!=".gr":
                print "den einai"
```

ΑΣΚΗΣΗ 2

```
EM=raw_input("Dose leksi")
c=0
while EM[-1]  not in "Ωω":
   if EM[0]=="A" or EM[0]=="α":
      c+=1
   EM=raw_input("Dose leksi")

print c
```

ΑΣΚΗΣΗ 3

```
EM=raw_input("Dose leksi")
S=""
while  EM != ".":
   S = S + EM + " "
   EM=raw_input("Dose leksi")
leksi = S[ : len(S)-1] +  "."
print leksi
```

ΑΣΚΗΣΗ 4

```
A = raw_input('δώσε μια λέξη: ')
F = True
N = len(A)
i = 0
while i<N/2 and F==True :
   if  A[i] != A[N-i-1] :
      F = False
   i += 1

if F==True:
   print "Einai"
else:
   print "Den einai"
```

ΑΣΚΗΣΗ 5

```
A=["Athens", "Paris", "London"]
c=0
for X in A:
   A[c]= 'W'+X[1:  ]
   c+=1
```

```
A=["Athens", "Paris", "London"]
for x  in range(len(A)):
        Y=A[x]
        A[x]= 'W'+Y[1:  ]
```

ΑΣΚΗΣΗ 6

```
A=[]

for  x in range(25):
   On=str(raw_input('Dose onoma'))
   A.append(On)
max1=0
y=""
for x in A:
   if len(x)>max1:
      max1=len(x)
      y=x
print y

F=False
for x in range(len(A)-1):
   if A[x] in A[x+1 : ]:
      F=True
      Synonimo=A[x]

if F==True:
   print "Den Yparxei synonimia"
else:
   print "Yparxei synonimia kai einai", Synonimo

max1=0
for x in A:
   c=0
   for i  in x:
      if i=="a" or i=="A":
         c=c+1
   if c>max1:
      max1=c
      y=x
print y

for X in A:
   if  X[len(X)-4 : ] =="akis":
      print X
```

ΑΣΚΗΣΗ 7

```
A=[]

B=str(raw_input('Dose symvoloseira'))

for x in B:
   A.insert(0,x)

print A
```

ΑΣΚΗΣΗ 8

```
B=str(raw_input('Dose symvoloseira'))
F="AEOIHYaeoihy"
c=0
for x in B:
   if  x  not in  F:
      c=c+1

print c
```

ΑΣΚΗΣΗ 9

```
def  find_New(A, c):

   for x in range(len(A)):
      if A[x] == c:
         return x
```

ΑΣΚΗΣΗ 10

```
def  count_New(A, c):
   z=0
   for x in  A :
      if x == c:
         z=z+1
   return  z
```

1) Να γίνει πρόγραμμα σε Python που:
Α) Ζητάει από τον χρήστη ένα κείμενο (συμβολοσειρά) που αποτελείται από λέξεις.
Β) Στη συνέχεια οι λέξεις μεταφέρονται μία μία σε μία λίστα. Θεωρούμε ότι μία λέξη τελείωσε όταν εντοπιστεί το κενό διάστημα " " ή ένας από τους χαρακτήρες ! , . : ; ?
Γ) Να εμφανιστούν μία μία οι λέξεις που μεταφέρθηκαν στη λίστα.

Παρατήρηση: Οι χαρακτήρες ! , . : ; ? και το κενό διάστημα " " δεν μεταφέρονται στη λίστα, ακόμα κι αν υπάρχουν στη σειρά.

2) Να γράψετε μια συνάρτηση isSubstring(string, substring) η οποία θα ελέγχει, αν η συμβολοσειρά substring περιέχεται στην συμβολοσειρά string και αν ναι, θα επιστρέφει True. Στη συνέχεια να χρησιμοποιήσετε αυτή τη συνάρτηση, για να βρείτε πόσες φορές εμφανίζεται μια συμβολοσειρά μέσα σε μια άλλη.

3) Να δίνετε από το πληκτρολόγιο έναν νέο κωδικό σε έναν λογαριασμό ηλεκτρονικού ταχυδρομείου και να γίνεται έλεγχος αν ο κωδικός είναι ασφαλής πληρώντας τις εξής προϋποθέσεις:

Α) Να περιέχει τουλάχιστον 8 χαρακτήρες

Β) Να περιέχει τουλάχιστον έναν κεφαλαίο χαρακτήρα

Γ) Να περιέχει κάποιον από τους χαρακτήρες @$!+&#

Δ) Να περιέχει γράμματα και αριθμούς

ΑΣΚΗΣΗ 1

```
T=raw_input('Δώσε ένα κέμενο.')
A=[]
s=''
for x in T:
   if x not in '!,.:;? ':
      s=s+x
   else:
      if s!='':
         A.append(s)
         s=''
for x in A:
   print x
```

ΑΣΚΗΣΗ 2

```
def isSubstring(S, S1):
   if S in S1:
      c=0
      N=len(S)
      for x in range(0,len(S1)-N+1,1):
         if   S1[x : x+N] == S:
            c=c+1

      print c

      return True


S="και"
S1="καλο**και**ρι **και** χεινώνα είναι καλο **και** το πο**και**μον"
print    isSubstring(S, S1)
```

ΑΣΚΗΣΗ 3

```
import string
L1=list( string.ascii_lowercase.upper())
L2=list( string.ascii_lowercase)
N=list("0123456789")
SB="@$!+&#"
F1=False
F2=False
F3=False
F4=False

S=raw_input("Δώσε κωδικό")

for x in S:
   if x in L1 :
      F1=True
   if x in N :
      F2=True
   if x in SB :
      F3=True
   if x in L2 :
      F4=True

if len(S)>=8 and F1==True and F2==True and F3==True and F4==True :
   print "Κατάλληλος κωδικός"
else:
  print "Ακατάλληλος κωδικός"
```

1) Σας δίνονται δύο λίστες:

list1 = ["M", "na", "i", "Ke"]

list2 = ["y", "me", "s", "lly"]

Γράψτε κώδικα ώστε να έχετε το παρακάτω αποτέλεσμα :

['My', 'name', 'is', 'Kelly']

2) Σας δίνεται η λίστα **aList = [1, 2, 3, 4, 5, 6, 7]**. Γράψτε κώδικα ώστε να έχετε το παρακάτω αποτέλεσμα :

[1, 4, 9, 16, 25, 36, 49]

3) Σας δίνονται δύο λίστες:

A = ["Hello ", "take "]

B = ["Dear", "Sir", "Bob"]

Γράψτε κώδικα ώστε να έχετε το παρακάτω αποτέλεσμα :

['Hello Dear', 'Hello Sir', 'Hello Bob', 'take Dear', 'take Sir', 'take Bob']

4) Να δίνετε από το πληκτρολόγιο μια συμβολοσειρά και αν δεν είναι καρκινική να τυπώνετε αντίστοιχο μήνυμα. Ένα καρκινικό κείμενο διαβάζεται ανάποδα Π.χ. τα ονόματα "ΑΝΝΑ" ή "ΣΑΒΒΑΣ" αλλά και πόλεις όπως "ΣΕΡΡΕΣ" είναι καρκινικές.

5) Να δίνετε από το πληκτρολόγιο μια συμβολοσειρά και να γράψετε κώδικα που θα μετρά και θα τυπώνει τον αριθμό των λέξεων της

6) Να δίνετε από το πληκτρολόγιο μια συμβολοσειρά και να γράψετε κώδικα που θα τυπώνει την συμβολοσειρά με γράμματα που θα εναλλάσσονται σε κεφαλαία και μικρά π.χ Η παπαρούνα θα γίνει ΠαΠαΡοΥνΑ

ΑΣΚΗΣΗ 1	**ΑΣΚΗΣΗ 2**

list1 = ["M", "na", "i", "Ke"]
list2 = ["y", "me", "s", "lly"]
list3=[]
for x in range(len(list1)):
  list3.append(list1[x]+list2[x])
print list3
	aList = [1, 2, 3, 4, 5, 6, 7]
	bList=[]
	for x in aList:
	  bList.append(pow(x,2))
	print bList

ΑΣΚΗΣΗ 3	**ΑΣΚΗΣΗ 4**

A = ["Hello ", "take "]
B = ["Dear", "Sir", "Bob"]
C=[]
for x in range(len(A)):
  for y in range(len(B)):
  C.append(A[x]+B[y])
print C
	A=raw_input("Dose protasi")
	if len(A)%2==0:
	  for x in range(len(A)/2):
	  if A[x]!=A[len(A)-x-1]:
	  print "Den einai"
	else:
	  print "Den einai"

ΑΣΚΗΣΗ 5	**ΑΣΚΗΣΗ 6**

A=raw_input("Dose protasi")
c=0
for x in A:
  if x==" ":
  c=c+1
 	
print "lekseis =", c+1
	A=raw_input("Dose leksi")
	S=""
	c=1
	for x in A:
	
	  if c%2==1:
	  S=S+x.upper()
	  else:
	  S=S+x
	  c=c+1
	print S

1) Ένα υποκατάστημα σούπερ μάρκετ έχει δημιουργήσει δύο λίστες για τα προϊόντα που πουλάει. Στη πρώτη λίστα, με όνομα bcodes, είναι οι γραμμωτοί κωδικοί των προϊόντων (bar codes) σε μορφή συμβολοσειράς και στη δεύτερη, με όνομα sales, είναι οι αντίστοιχες συνολικές πωλήσεις των προϊόντων σε ευρώ (€) για το έτος 2020.

Π.χ.
bcodes [‘5201171000126’,’6901100341234’,…….,’8300112291033’],
sales [1253,3872,……..,7812]

Οι δύο λίστες έχουν ακριβώς τις ίδιες θέσεις.
Οι αντίστοιχες θέσεις της κάθε λίστας αναφέρονται στο ίδιο προϊόν.

Ο διευθυντής του υποκαταστήματος ζήτησε από τον προγραμματιστή να δημιουργήσει μια συνάρτηση σε python, με παραμέτρους τις δύο λίστες (bcodes,sales), που να αναζητά στην πρώτη λίστα τα ελληνικά προϊόντα, γνωρίζοντας ότι οι ελληνικοί γραμμωτοί κωδικοί ξεκινούν από ‘520’, και να βάζει τους ελληνικούς κωδικούς σε μια λίστα grcodes και τις αντίστοιχες πωλήσεις σε μια λίστα grsales.
Στη συνέχεια θα ταξινομεί τις δύο λίστες grcodes και grsales σε φθίνουσα σειρά (από τις μεγαλύτερες προς τις μικρότερες) σύμφωνα με τις πωλήσεις και θα τις επιστρέφει (return) σε μορφή λίστας με δύο υπολίστες [grcodes,grsales].

ΑΣΚΗΣΗ 1

```
def greek(bcodes,sales):

   N=len(bcodes)
   grcodes=[ ]
   grsales=[ ]

   for  x  in  range(N):
      B= bcodes[x]
      if  B[ : 3 ]=='520':
         grcodes.append(bcodes[x])
         grsales.append(sales[i])

  N=len(grsales)

   for i in range(1,N,1):
      for j in range(N-1, i-1,-1):
         if grsales[j]>grsales[j-1]:
            grsales[j],grsales[j-1]=grsales[j-1],grsales[j]
            grcodes[j],grcodes[j-1]=grcodes[j-1],grcodes[j]

   return [grcodes,grsales]

bbcodes=['520000001', '46733322','52756566', '13456777', '456889900', '5206565789', '67898745433',
'520000786', '89796908797', '797767865']

ssales=[12346,565477, 312346, 886868, 335465, 878787, 24335, 1313131, 898989, 907878]

R=greek(bbcodes, ssales)
for  x  in  R:
   print x
```

1) Γράψτε πρόγραμμα σε Python που να δημιουργεί μια λίστα, η οποία θα περιέχει τους άρτιους αριθμούς (ζυγούς) από το 1 έως το 50. Στη συνέχεια το πρόγραμμα θα εκτυπώνει τη λίστα και το μήκος της.

2) Να γράψετε πρόγραμμα σε python το οποίο θα δημιουργεί μια λίστα με όλα τα πολλαπλάσια του 5 από το 10 έως το 100. Στην συνέχεια θα εκτυπώνει την λίστα αντίστροφα δηλ. πρώτα το 100 μετά το 95 κ.ο.κ.

3) Γράψτε πρόγραμμα σε Python που να διαβάζει τα ονόματα 10 μαθητών και τους βαθμούς τους στο Προγραμματισμό. Το πρόγραμμα θα υπολογίζει και θα εμφανίζει το ποσοστό των μαθητών που έχουν βαθμό πάνω από 18 (έλεγχος ορθότητας τιμών βαθών μεταξύ 0 και 20)

4) Να δημιουργήσετε μια λίστα η οποία θα περιλαμβάνει τα ονόματα των ημερών της εβδομάδας. Στην συνέχεια να διαγράψετε το Σάββατο και την Κυριακή. Να εισάγετε στην αρχή της λίστας την Κυριακή και στο τέλος της λίστας το Σάββατο. Να τυπώσετε πάλι την λίστα.

ΑΣΚΗΣΗ 1

```
A=[ ]
for x in range( 2, 51, 2):
        A.append(x)
print A, len(A)
```

ΑΣΚΗΣΗ 2

```
A=[ ]
for x in range( 10, 101, 5):
        A.append(x)
for x in range( 100, 9, -5):
        print x
```

ΑΣΚΗΣΗ 3

```
C=0
for x in range( 10):
        V = input ('dose vathmo')
        While V<0 or V > 20:
                V = input ('dose sosto vathmo')
        If  V>18:
                C=C+1
Pososto= 100 * C /10
```

ΑΣΚΗΣΗ 4

```
A=['Δευτέρα', 'Τρίτη', 'Τετάρτη', 'Πέμπτη', 'Παρασκευή', 'Σάββατο', 'Κυριακή']
A.pop(5)
A.pop(6)
A.insert(0,'Κυριακή')
A.append('Σάββατο')
print A
```

1) Μια εταιρεία Πληροφορικής καταγράφει, για 3 ιστότοπους, τον αριθμό των επισκέψεων που δέχεται ο καθένας, κάθε μέρα, για 30 ημέρες.

Να αναπτύξετε αλγόριθμο, ο οποίος:

Δ1. Για καθένα από τους ιστότοπους να διαβάζει τον αριθμό των επισκέψεων που δέχθηκε ο ιστότοπος για καθεμιά ημέρα.

Δ2. Να εμφανίζει για κάθε ιστότοπο τον συνολικό αριθμό των επισκέψεων που δέχθηκε αυτός στο διάστημα των 30 ημερών.

Δ3. Να εμφανίζει για κάθε ιστότοπο το μήνυμα «επιτυχία» αν δέχθηκαν περισσότερες από 500 επισκέψεις κάθε ημέρα, στο διάστημα των 30 ημερών αλλιώς να εμφανίζει το μήνυμα «αποτυχία»

Δ4. Να ταξινομήσετε και τους τρείς ιστότοπους με βάση την μέθοδο της φυσαλίδας.

ΑΣΚΗΣΗ 1

```
def   epis(A):
        S=0
        for  x in A:
                S=S+x
        return  S

def  Epit (A):
        T=True
        for x in A:
                if x<=500:
                T=False
        if T==True:
                return "Epitixia"
        else:
                return  "Apotixia"

Isto1=[]
Isto2=[]
Isto3=[]

for j in range(30):
   x=input('dose episkepsimotita Isto1')
   Isto1.append(x)
   Isto2.append(input('dose episkepsimotita Isto2'))
   Isto3.append(input('dose episkepsimotita Isto3'))

print "Synolo Istos1 =", epis(Isto1)
print "Synolo Istos2 =", epis(Isto2)
print "Synolo Istos3 =", epis(Isto3)

print  “Gia ton istotopo 1 “, Epit (Isto1)
print  “Gia ton istotopo 2 “, Epit (Isto2)
print  “Gia ton istotopo 3 “, Epit (Isto3)

for i in range (1,30,1):
   for j in range (30-1, i-1, -1):
      if Isto1[j]<Isto1[j-1]:
         Isto1[j], Isto1[j-1]=Isto1[j-1], Isto1[j]
      if Isto2[j]<Isto2[j-1]:
         Isto2[j], Isto2[j-1]=Isto2[j-1], Isto2[j]
      if Isto3[j]<Isto3[j-1]:
         Isto3[j], Isto3[j-1]=Isto3[j-1], Isto3[j]
```

1) Να γράψετε πρόγραμμα σε Python το οποίο:

α) να διαβάζει το επώνυμο και το μισθό 30 υπαλλήλων μιας εταιρείας και να τα καταχωρίζει στις λίστες Ε και Μ αντίστοιχα

β) να υπολογίζει και να εμφανίζει το ΜΟ όλων των μισθών

γ) να υπολογίζει και να εμφανίζει το μέγιστο μισθό και το επώνυμο του υπαλλήλου που τον έχει

δ) με τη βοήθεια της bubblesort να ταξινομεί τις λίστες με αύξουσα σειρά ως προς το μισθό και να εμφανίζει στην οθόνη όλους τους υπαλλήλους με το μισθό τους από δίπλα (σε αύξουσα σειρά)

ε) να εμφανίζει το επώνυμο των υπαλλήλων που έχουν μισθό πάνω από το ΜΟ

στ) να εμφανίζει το επώνυμο και το μισθό των τριών υπαλλήλων με το μικρότερο μισθό (θεωρούμε ότι είναι μόνο 3)

2) Να γράψετε πρόγραμμα σε Python το οποίο:

α) να διαβάζει τους αριθμούς μητρώου και τους βαθμούς 20 μαθητών στο μάθημα της ιστορίας και να τους καταχωρίζει σε δύο λίστες ΑΜ και ΒΑ αντίστοιχα

β) να υπολογίζει και να εμφανίζει το ΜΟ όλων των βαθμών των μαθητών

γ) να υπολογίζει και να εμφανίζει πόσοι μαθητές είχαν βαθμό κάτω από το ΜΟ της τάξης

δ) να υπολογίζει και να εμφανίζει μία λίστα κατά αύξουσα σειρά ως προς το βαθμό, που να έχει τον ΑΜ και από δίπλα το βαθμό του κάθε μαθητή. (με τη βοήθεια της bubblesort)

ε) Θεωρώντας ότι μόνο ένας μαθητής έχει το βαθμό 14 με τη βοήθεια της binarysearch που επιστρέφει τη θέση να βρείτε τον ΑΜ του μαθητή με αυτό το βαθμό.

ΑΣΚΗΣΗ 1

α) να διαβάζει το επώνυμο και το μισθό 30 υπαλλήλων μιας εταιρείας και να τα καταχωρίζει στις λίστες Ε και Μ αντίστοιχα

```
E=[]
M=[]
for  x  in range(30):
   Epo=str(raw_input('Dose eponimo'))
   Mis=input("Dose mistho")
   E.append(Epo)
   M.append(Mis)
```

β) να υπολογίζει και να εμφανίζει το ΜΟ όλων των μισθών

```
S=0.0
for  x  in M:
   S=S+x
MO=S/30
print "Mosos Oros Misthon = ", MO
```

γ) να υπολογίζει και να εμφανίζει το μέγιστο μισθό και το επώνυμο του υπαλλήλου που τον έχει

```
Max1=M[0]
y=0
for  x  in range(1,len(M),1):
   if  M[x]>Max1:
      Max1=M[x]
      y=x

print "O ", E[y], "exei ton megal. mistho iso me ", M[y]
```

```
Max1=0
y=0
for  x  in range(len(M)):
   if  M[x]>Max1:
      Max1=M[x]
      y=x

print "O ", E[y], "exei ton megal. mistho iso me ", M[y]
```

δ) με τη βοήθεια της bubblesort να ταξινομεί τις λίστες με αύξουσα σειρά ως προς το μισθό και να εμφανίζει στην οθόνη όλους τους υπαλλήλους με το μισθό τους από δίπλα (σε αύξουσα σειρά)

```
N=30
for  i  in range(1,N,1):
   for  j in range(N-1,i-1,-1):
      if M[j]<M[j-1]:
         M[j],M[j-1]=M[j-1],M[j]
         E[j],E[j-1]=E[j-1],E[j]

for x in range(30):
   print E[x], M[x]
```

ε) να εμφανίζει τα επώνυμα των υπαλλήλων που έχουν μισθό πάνω από το ΜΟ

```
for x in range(30):
   if M[x]>MO:
      print   E[x]
```

στ) να εμφανίζει το επώνυμο και το μισθό των τριών υπαλλήλων με το μικρότερο μισθό (θεωρούμε ότι είναι μόνο 3)

```
print E[0], M[0]
print E[1], M[1]
print E[2], M[2]
```

ΑΣΚΗΣΗ 2

```
def    binarysearch( BA, v ) :
   first = 0

   last = len(BA)-1
   found = False
   while   found== False :

      mid = ( first + last ) // 2
      if    BA[ mid ] == v :
         found = True
      elif   BA[ mid ] < v :
         first = mid + 1
      else:
         last = mid-1
   return   mid

AM=[]
BA=[]
for  x  in range(20):
   am=str(raw_input('Dose AM'))
   v=input("Dose Vathmo")
   AM.append(am)
   BA.append(v)

S=0.0
for  x  in BA:
   S=S+x

MO=S/20

print "Mosos Oros Misthon = ", MO

c=0
for x in range(20):
   if BA[x]<MO:
      c=c+1
print c

N=20
for  i  in range(1,N,1):
   for  j in range(N-1,i-1,-1):
      if BA[j]<BA[j-1]:
         BA[j],BA[j-1]=BA[j-1],BA[j]
         AM[j],AM[j-1]=AM[j-1],AM[j]

ALL=[]

for x in range(20):
   ALL.append(AM[x] + " " + str(BA[x]))

print   ALL
print   AM[ binarysearch( BA, 14 ) ]
```

1) Σε ένα αρχείο “poleis.txt” έχουμε γραμμένα ονόματα πόλεων, ένα σε κάθε γραμμή. Να γράψετε πρόγραμμα σε python το οποίο:

α) να διαβάζει κάθε πόλη από το αρχείο και να την καταχωρίζει σε μία λίστα POL

β) να εμφανίζει στην οθόνη του υπολογιστή τις πόλεις με την αντίστροφη σειρά από αυτή που διαβάστηκαν.

γ) να υπολογίζει και να εμφανίζει την πόλη με τα λιγότερα γράμματα (θεωρώντας ότι είναι μοναδική)

δ) να υπολογίζει και να εμφανίζει την πόλη με τα περισσότερα γράμματα (Θεωρώντας ότι είναι μοναδική)

ε) να υπολογίζει και να εμφανίζει πόσες πόλεις έχουν μέσα το γράμμα “Τ” ή το “t”

2) Σε ένα ΕΠΑΛ οι 20 μαθητές έγραψαν διαγώνισμα στο μάθημα του Προγραμματισμού και ο καθηγητής τους το διόρθωσε με άριστα το 100. Να γράψετε πρόγραμμα σε Python το οποίο:

α) να διαβάζει για κάθε μαθητή το επώνυμο, το όνομά του και το βαθμό που έγραψε.

β) να κάνει έλεγχο ορθότητας στο βαθμό ώστε να είναι από 0 μέχρι και 100

γ) να καταχωρίζει το επώνυμο σε μία λίστα ΕΡ , το όνομα σε μία λίστα ΟΝ και το βαθμό σε μία λίστα ΒΑ

δ) να υπολογίζει και να εμφανίζει το ΜΟ όλου του τμήματος

ε) να εμφανίζει τα ονοματεπώνυμα των μαθητών που πήραν πάνω από το ΜΟ της τάξης.

στ) να υπολογίζει τον μεγαλύτερο βαθμό της τάξης

ζ) να εμφανίζει τα ονοματεπώνυμα των μαθητών που πήραν αυτό το βαθμό.

ΑΣΚΗΣΗ 1

```
POL=[]

f1=open("poleis.txt", "r")

for x in f1:
   POL.insert( 0, x[0 : len(x)-1 ] )

f1.close()

for x in POL:
   print x

max1=len(POL[0])
min1=len(POL[0])

y=POL[0]
z=POL[0]

for x in range(1,len(POL),1):

   if len(POL[x])<min1:
      min1=len(POL[x])
      y=POL[x]

   if len(POL[x])>max1:
      max1=len(POL[x])
      z=POL[x]

print y
print z

c=0
for x in POL:
   if  'T' in x or 't' in x:
      c=c+1
print c
```

ΑΣΚΗΣΗ 2

```
EP=[]
ON=[]
BA=[]

for  x  in range(20):
   E=str(raw_input('Dose Eponimo'))
   O=str(raw_input('Dose Onoma'))
   B=input('Dose vathmo')
   while B<0 or B>100:
      B=input('Dose SOSTO vathmo')

   EP.append(E)
   ON.append(O)
   BA.append(B)

S=0.0
for x in BA:
   S=S+x

MO=S/20

for x  in range(len(BA)):
   if BA[x]>MO:
      print EP[x], ON[x]

max1=0

for x  in range(len(BA)):
   if BA[x]>max1:
      max1=BA[x]

for x  in range(len(BA)):
   if BA[x]==max1:
      print EP[x], ON[x]
```

1) Να αναπτύξετε αλγόριθμο ο οποίος:

α. θα διαβάζει τον αριθμό των μαθητών της Α' λυκείου ενός σχολείου. Πρέπει να είναι θετικός αριθμός.

β. θα διαβάζει για κάθε μαθητή το όνομά του και το βαθμό του (στην εικοσαβάθμια κλίμακα) πραγματοποιώντας έλεγχο δεδομένων.

γ. θα εκτυπώνει πόσες λάθος καταχωρήσεις (λάθος βαθμοί) δόθηκαν.

δ. θα εκτυπώνει το μέσο όρο βαθμολογίας της τάξης.

ε. θα εκτυπώνει τα ονόματα των μαθητών που έχουν βαθμό πλησιέστερα στο μέσο όρο.

2) Να γράψετε πρόγραμμα που θα χρησιμοποιεί τη λίστα A, στην οποία μπορεί να τοποθετήσει το πολύ 50 στοιχεία, και θα εκτελεί τις παρακάτω ενέργειες.

i. Θα διαβάζει έναν αριθμό που θα καθορίζει το είδος της ενέργειας («1» για εισαγωγή αριθμού στη στοίβα, «2» για εξαγωγή και «3» για τερματισμό της επανάληψης).

ii. Κάθε ενέργεια θα εκτελείται εφόσον αυτό είναι δυνατόν. Έτσι αν δεν μπορεί να γίνει εισαγωγή θα εμφανίζεται το μήνυμα «Γεμάτη στοίβα», ενώ αν δεν μπορεί να γίνει εξαγωγή το μήνυμα «Άδεια στοίβα».

iii. Μετά το τέλος της επαναληπτικής διαδικασίας θα εμφανίζει πόσες φορές άδειασε και πόσες φορές γέμισε η στοίβα καθώς και το πλήθος των στοιχείων που έχει.

3) Ένας χορευτικός σύλλογος έχει μέλη άντρες και γυναίκες. Πρόκειται να κάνει μια εκδήλωση με επίδειξη διάφορων χορών, μερικοί από τους οποίους χορεύονται σε ζεύγη που αποτελούνται από έναν άντρα και μία γυναίκα. Επειδή αυτοί οι χοροί προϋποθέτουν κάποια εμπειρία μεγαλύτερη από τους άλλους, ο σύλλογος θα δώσει προτεραιότητα στα αρχαιότερα μέλη του συλλόγου που μπορούν να φτιάξουν ζεύγη, με στόχο όμως να φτιαχτούν όσα περισσότερα ζευγάρια είναι δυνατό.

Να γράψετε πρόγραμμα που:

α) Θα διαβάζει το πλήθος των μελών του συλλόγου ελέγχοντας ώστε να είναι θετικός αριθμός και το πολύ ίσος με 100.

Θα διαβάζει και θα τοποθετεί στη λίστα ON τα ονοματεπώνυμα των μελών του συλλόγου και σε μια παράλληλη με αυτήν λίστα FILO το φύλο Α ή W του κάθε μέλους, για άντρα ή γυναίκα αντίστοιχα, κάνοντας τον απαιτούμενο έλεγχο εγκυρότητας για το φύλο.

Θεωρείστε δεδομένο ότι τα στοιχεία των μελών εισάγονται με σειρά αρχαιότητας συμμετοχής στο σύλλογο.

β) Στη συνέχεια το πρόγραμμα θα δημιουργεί δύο στοίβες Μ και W, όπου χρησιμοποιώντας επαναληπτικά τη λειτουργία της ώθησης, στην πρώτη στοίβα θα ωθήσει τα ονόματα όλων των ανδρών του συλλόγου και στη δεύτερη όλα τα ονόματα των γυναικών του συλλόγου. Η ώθηση των ονομάτων στις δύο στοίβες θα πρέπει να γίνεται ώστε σε κάθε στοίβα τα ονοματεπώνυμα να υπάρχουν με σειρά αρχαιότητας στο σύλλογο από την κορυφή της στοίβας και διαδοχικά μέχρι το τέλος της, δηλαδή το αρχαιότερο μέλος θα βρίσκεται στην κορυφή κάθε στοίβας ενώ το νεώτερο στο τέλος της.

γ) Χρησιμοποιώντας επαναληπτικά τη λειτουργία της απώθησης στις δύο στοίβες θα εμφανίζει τα ονοματεπώνυμα των ζευγαριών που θα χορέψουν στην εκδήλωση. Τα ζευγάρια θα δημιουργούνται με σειρά αρχαιότητας, δηλαδή ο αρχαιότερος χορευτής θα γίνεται ζευγάρι με την αρχαιότερη χορεύτρια, ο αμέσως επόμενος αρχαιότερος με την αμέσως επόμενη αρχαιότερη κ.ο.κ.

δ) Θα εμφανίζει επίσης πόσα ζευγάρια φτιάχτηκαν για να χορέψουν τελικά.

4) Μία αεροπορική εταιρία εκτελεί το δρομολόγιο Θεσσαλονίκη – Χανιά κατά τους θερινούς μήνες του έτους. Λόγω της αυξανόμενης ζήτησης, η εταιρία διατηρεί σε ουρά αναμονής τους επιβάτες που δεν πρόλαβαν να κλείσουν εισιτήριο ώστε αν προκύψει κάποια ακύρωση τότε να ενημερώσει τον πρώτο στη σειρά πελάτη που εισήχθη στην λίστα αναμονής να κλείσει εισιτήριο. Η λίστα αναμονής δεν μπορεί να περιλαμβάνει περισσότερα από 10 ονόματα.

Να κάνετε πρόγραμμα το οποίο:

α. Θα δέχεται μία εκ των τριών τιμών: «ΕΓΓΡΑΦΗ», «ΑΚΥΡΩΣΗ» ή «ΤΕΛΟΣ», σαν επιλογή του χρήστη κάνοντας έλεγχος εγκυρότητας.

β. Αν ο χρήστης δώσει την τιμή «ΕΓΓΡΑΦΗ» τότε θα ζητείται το όνομα του χρήστη και θα καταχωρείται στην λίστα αναμονής μόνο εφόσον η λίστα αναμονής είναι μικρότερη ή ίση των 10 ατόμων. Διαφορετικά να εμφανίζει το μήνυμα: «Η λίστα αναμονής είναι πλήρης».

γ. Αν ο χρήστης δώσει την τιμή «ΑΚΥΡΩΣΗ», τότε κάποιος από τους επιβάτες της πτήσης έχει ακυρώσει την κράτησή του, συνεπώς το πρόγραμμα θα πρέπει να εμφανίσει το όνομα του ατόμου που είναι πρώτο διαθέσιμο στην λίστα αναμονής **και να το διαγράψει από την λίστα**.

Αν δεν υπάρχουν άτομα στην λίστα αναμονής, να εμφανίζεται το μήνυμα «Η λίστα αναμονής είναι άδεια».

δ. Η παραπάνω διαδικασία να επαναλαμβάνεται μέχρι ο χρήστης να δώσει την τιμή «ΤΕΛΟΣ».

ε. Το πρόγραμμα να εμφανίζει το πλήθος των ατόμων που κατάφεραν να κάνουν κράτηση μέσα από την λίστα αναμονής, καθώς και το μέγιστο πλήθος των ατόμων που περίμεναν στην ουρά αναμονής.

ΑΣΚΗΣΗ 1

```
N=input("Δωσε αριθμό μαθητών")
while N<=0:
   N=input("Δωσε ΣΩΣΤΟ αριθμό μαθητών")

c=0

A=[]
ON=[]
for x in range(N):
   on=str(raw_input("Δωσε όνομα"))
   V=input("Δωσε τον βαθμό")

   while V<0 or V>20:
      V=input("Δωσε ΣΩΣΤΟ βαθμό")
      c=c+1

   ON.append(on)
   A.append(V)

S=0.0
for x  in A:
   S=S+x

MO=S/len(A)

min1=21
min2=21
for x in range(len(A)):
   if A[x] <= MO :
      if MO-A[x] < min1 :
         min1=MO-A[x]
         Z1=x
   else:
      if abs(MO-A[x]) < min2 :
         min2=adb(MO-A[x])
         Z2=x

for x in range(len(A)):
   if A[x]==A[Z1] or A[x]==A[Z2]:
      print ON[x]

print "δόθηκαν λάθος βαθμοί ", c

print "μέσος όρος βαθμολογίας  ", MO
```

ΑΣΚΗΣΗ 2

```
c1=0
c2=0
A=[]

F1=False
F2=False
E=input("1 εισαγωγή 2 εξαγωγή 3 τέλος")
while E!=3:

   if E==1:
      if len(A)<50:
         x=input("Δωσε στοιχείο")
         A.append(x)
         if F1==True:
            F1=False
      else:
         print "Γεμάτη στοίβα"
   if E==2:
      if len(A)!=0:
         A.pop()
         if F2==True:
            F2=False
      else:
         print "Άδεια στοίβα"

   if len(A)==50 and F1!=True:
      c1=c1+1
      F1=True
   if len(A)==0 and F2!=True:
      c2=c2+1
      F2=True

   E=input("1 εισαγωγή 2 εξαγωγή 3 τέλος")

print "Γέμισε ", c1, "φορές"
print "Άδειασε  ", c2, "φορές"
print "Πλήθος στοιχείων ", len(A)
```

ΑΣΚΗΣΗ 3

```
ON=[]
FILO=[]
M=[]
W=[]

N=input("Δώσε αριθμό μελών")
while N<0 or N>100:
   N=input("Δώσε ΣΩΣΤΟ αριθμό μελών")


for x in range(N):
   on=str(raw_input("Δώσε ονομα"))
   filo=str(raw_input("Δώσε φύλο"))
   ON.append(on)
   FILO.append(filo)

for x in rnage(len(ON)):
   if FILO[x]=="M":
      M.append(ON[x])
   if FILO[x]=="W":
      W.append(ON[x])
c=0
while M!=[] and W!=[] :
   print M[0], W[0]
   M.pop(0)
   W.pop(0)
   c=c+1

print "ζευγάρια ", c
```

ΑΣΚΗΣΗ 4

```
A=[]
c1=0
c2=0

E=raw_input("ΕΓΓΡΑΦΗ ή ΑΚΥΡΩΣΗ ή ΤΕΛΟΣ")
while E not in ["ΕΓΓΡΑΦΗ", "ΑΚΥΡΩΣΗ", "ΤΕΛΟΣ"] :
   E=raw_input("ΕΓΓΡΑΦΗ ή ΑΚΥΡΩΣΗ ή ΤΕΛΟΣ")

while E!="ΤΕΛΟΣ":

   if E=="ΕΓΓΡΑΦΗ" and len(A) < 10:
      on=str(raw_input("Δώσε ονομα"))
      A.append(on)
      c2=c2+1
   else:
      print "Η λίστα αναμονής είναι πλήρης"

   if E=="ΑΚΥΡΩΣΗ" and len(A) > 0:
      print A[0]
      A.pop(0)
      c1=c1+1
   else:
      print "Η λίστα αναμονής είναι άδεια"

   E=raw_input("ΕΓΓΡΑΦΗ ή ΑΚΥΡΩΣΗ ή ΤΕΛΟΣ")
   while E not in ["ΕΓΓΡΑΦΗ", "ΑΚΥΡΩΣΗ", "ΤΕΛΟΣ"] :
      E=raw_input("ΕΓΓΡΑΦΗ ή ΑΚΥΡΩΣΗ ή ΤΕΛΟΣ")

print "κατάφεραν να κάνουν κράτηση ", c1

print "περίμεναν στην ουρά αναμονής", c2
```

1) Στο πρωτάθλημα Μπάσκετ κάθε ομάδα παίρνει 2 βαθμούς για κάθε νίκη και ένα βαθμό για κάθε ήττα. Να γράψετε πρόγραμμα σε python το οποίο:

α) να διαβάζει το όνομα της ομάδας μέχρι να δοθεί για όνομα η λέξη " **TELOS**" και να το καταχωρεί σε μία λίστα **ON**

β) να διαβάζει τις νίκες και τις ήττες της ομάδας και να τις καταχωρεί στις λίστες **NIKES** και **HTTES** αντίστοιχα.

γ) να υπολογίζει και να εμφανίζει τη συνολική βαθμολογία που έχει η κάθε ομάδα

δ) να καταχωρεί τη συνολική βαθμολογία σε μία λίστα **SYN**

ε) με τη βοήθεια του αλγορίθμου **bubbleSort** (αλγόριθμου της φυσαλίδας) να ταξινομεί τις ομάδες (όλες τις λίστες) κατά **φθίνουσα** σειρά ως προς τη συνολική βαθμολογία.

στ) να εμφανίζει για κάθε ομάδα, το όνομα ,τις νίκες, τις ήττες και τη συνολική βαθμολογία κατά φθίνουσα σειρά ως προς τη συνολική βαθμολογία.

ζ) να εμφανίζει τα ονόματα των ομάδων που έχουν κάτω από 4 ήττες.

η) να δημιουργεί ένα **αρχείο** με το όνομα "apotelesmata.txt" και μέσα να εγγράφει όλα όσα εμφανίζονται στο βήμα στ)

2) Μία ερευνητική ομάδα πανεπιστημίου καταγράφει κάθε μία ώρα τη θερμοκρασία σε μία συγκεκριμένη περιοχή. Αυτό συμβαίνει για όλες τις ώρες της ημέρας. Να γράψετε πρόγραμμα σε Python το οποίο:

α) να δημιουργεί μία λίστα **WRES** η οποία να περιέχει τις τιμές από 1 μέχρι και 24 (οι ώρες της ημέρας) [1,2,...24]

β) να διαβάζει 24 θερμοκρασίες με τη σειρά που υπάρχουν οι ώρες στην λίστα WRES. Δηλαδή πρώτα τη θερμοκρασία για την ώρα 1 μετά για την ώρα 2 μέχρι και τη θερμοκρασία για την ώρα 24.

γ) να τις καταχωρίζει στη λίστα **THERM**

δ) να υπολογίζει και να εμφανίζει τη μέση θερμοκρασία της ημέρας

ε) να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της **ευθείας ανταλλαγής** (φυσαλίδα-bubble sort) τις δύο λίστες σε **φθίνουσα** σειρά ως προς τις θερμοκρασίες και να τις εμφανίζει με αυτή τη σειρά, εμφανίζοντας για κάθε ώρα το μήνυμα: "Την ώρα Χ είχαμε θερμοκρασία Υ"

στ) να υπολογίζει και να εμφανίζει τις τρεις μεγαλύτερες θερμοκρασίες καθώς επίσης και τις ώρες που αυτές επιτεύχθηκαν.

3) Να γράψετε πρόγραμμα σε Python το οποίο:

α) να διαβάζει το επώνυμο και το όνομα των μαθητών μιας τάξης και το σταθερό τηλέφωνο του καθενός. Τα στοιχεία αυτά καταχωρίζονται στις λίστες **EP,ON** και **THL** αντίστοιχα. Η διαδικασία αυτή να τερματίζεται όταν δοθεί ως Επώνυμο η λέξη «**TELOS**».

β) στη συνέχεια να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της **ευθείας ανταλλαγής** τις τρεις λίστες σε **αύξουσα** σειρά ως προς το επώνυμο.

γ) στη συνέχεια να εμφανίζει το επώνυμο, το όνομα και το τηλέφωνο για όλους τους μαθητές με αύξουσα σειρά ως προς το επώνυμο.

δ) στη συνέχεια να γράψετε τη **συνάρτηση** def binarySearch(lista, key) η οποία να επιστρέφει τη θέση του key μέσα στο lista

ε) να διαβάζει ένα επώνυμο και με τη βοήθεια της παραπάνω (δ) συνάρτησης να εμφανίζει το σταθερό τηλέφωνο του μαθητή με αυτό το επώνυμο ή το μήνυμα “δεν υπάρχει” αν δεν υπάρχει αυτό το επώνυμο.

Υποδείξεις: Θεωρούμε ότι όλοι οι μαθητές έχουν διαφορετικά επώνυμα και ότι όλα τα τηλέφωνα ξεκινούν από αριθμό που δεν είναι 0.

στ) (Δύσκολο) Τροποποιήστε τη bubblesort ώστε αν είχαμε δύο μαθητές με το ίδιο επώνυμο να εξετάζαμε το όνομά τους και η κατάταξή τους να γινόταν με βάση το όνομα. Δηλαδή ο Αθανασιάδης Ιωάννης πρέπει να ήταν πριν από τον Αθανασιάδη Κωνσταντίνο.

ΑΣΚΗΣΗ 1

```
ON=[]
NIKES=[]
HTTES=[]
SYN=[]

onoma=str(raw_input('Dose onoma'))
while  onoma != 'TELOS':
   N=input('Dose nikes')
   H=input('Dose Httes')
   ON.append(onoma)
   NIKES.append(N)
   HTTES.append(H)

   onoma=str(raw_input('Dose onoma'))

for x in range(len(NIKES)):
   print ON[x], NIKES[x]+HTTES[x]
   SYN.append(NIKES[x]+HTTES[x])


N=len(SYN)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if SYN[j]>SYN[j-1]:
         SYN[j],SYN[j-1]=SYN[j-1],SYN[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         NIKES[j],NIKES[j-1]=NIKES[j-1],NIKES[j]
         HTTES[j],HTTES[j-1]=HTTES[j-1],HTTES[j]


for x in range(len(SYN)):
   print ON[x], NIKES[x], HTTES[x], SYN[x]

for x in range(len(HTTES)):
   if HTTES[x]<4:
      print ON[x]

f1=open("apotelesmata.txt", "w")

for x in range(len(SYN)):
   f1.write (str(ON[x]) + "\n")
   f1.write (str(NIKES[x]) + "\n")
   f1.write (str(HTTES[x]) + "\n")
   f1.write (str(SYN[x]) + "\n")
f1.close()
```

ΑΣΚΗΣΗ 2

```
WRES=[]

THERM=[]

for x in range(24):
   T=input('Dose thermokrasia')
   THERM.append(T)
   WRES.append(x+1)

S=0.0
for x  in TERM:
   S=S+x

MO=S/24
print MO

N=len(THERM)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if THERM[j]>THERM[j-1]:
         THERM[j],THERM[j-1]=THERM[j-1],THERM[j]
         WRES[j],WRES[j-1]=WRES[j-1],WRES[j]

for x in range(len(THERM)):
   print "Tin ", WRES[x], " ora eixame thermokrasia ", THERM[x]

print "Tin ", WRES[0], " ora eixame thermokrasia ", THERM[0]
print "Tin ", WRES[1], " ora eixame thermokrasia ", THERM[1]
print "Tin ", WRES[2], " ora eixame thermokrasia ", THERM[2]
```

ΑΣΚΗΣΗ 3

```
EP=[]
ON=[]
THL=[]

E=str(raw_input('Dose eponimo'))
while  E != 'TELOS':
   onoma=input('Dose onoma')
   T=input('Dose thlefono')
   ON.append(onoma)
   EP.append(E)
   THL.append(T)
   E=str(raw_input('Dose eponimo'))

N=len(EP)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if EP[j]<EP[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         THL[j],THL[j-1]=THL[j-1],THL[j]

for x in range(len(EP)):
   print EP[x], ON[x], THL[x]

eponimo=raw_input('Dose eponimo')
A=binarysearch( EP, eponimo )
if A==-1:
   print "Den yparxei"
else:
   print THL[A]
```

Α΄ τρόπος

```
N=len(EP)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if   EP[j]<EP[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         THL[j],THL[j-1]=THL[j-1],THL[j]

N=len(EP)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if   EP[j]==EP[j-1]   and   ON[j]<ON[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         THL[j],THL[j-1]=THL[j-1],THL[j]
```

Β΄ τρόπος

```
N=len(EP)

for i in range(1,N,1):

   for j in range(N-1,i-1,-1):

      if   EP[j]<EP[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         THL[j],THL[j-1]=THL[j-1],THL[j]

      if   EP[j]==EP[j-1]   and   ON[j]<ON[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         THL[j],THL[j-1]=THL[j-1],THL[j]
```

1) Να γραφεί πρόγραμμα το οποίο:

Α. Δημιουργεί 3 λίστες (Ο,Τ και Ε) για την αποθήκευση στοιχείων των 300 προϊόντων ενός καταστήματος. Τη λίστα Ο για τα ονόματα, την Τ για τις τιμές και την Ε για την εταιρεία που προμηθεύει κάθε προϊόν.

Β. Να διαβάζει το όνομα μιας προμηθεύτριας εταιρείας.

Γ. Αν η εταιρεία βρεθεί στην λίστα Ε να εμφανίζει όλα τα στοιχεία για τα προϊόντα που προμηθεύει σε αύξουσα σειρά ως προς την τιμή τους. Σε περίπτωση που κάποια προϊόντα έχουν την ίδια αξία τότε να εμφανίζεται πρώτο το προϊόν με το μικρότερο αλφαβητικά όνομα.

Δ. Τέλος αν δεν βρεθεί η εταιρεία που εισάγει ο χρήστης να εμφανίζεται το κατάλληλο μήνυμα.

2) Να γραφεί πρόγραμμα Python το οποίο:

Α. Να διαβάζει το πλήθος των παικτών που σημείωσαν τέρματα στο προηγούμενο πρωτάθλημα ποδοσφαίρου. Να γίνει έλεγχος ορθότητας εγκυρότητας των δεδομένων εισόδου ώστε ο αριθμός των παικτών να είναι θετικός και να μην ξεπερνά το εκατό.

Β. Να διαβάζει και να τοποθετεί σε λίστες με ονόματα name και goal αντίστοιχα, τα ονόματα και τα τέρματα που σημείωσαν οι παίκτες.

Γ. Να ταξινομεί τις λίστες σύμφωνα με τα τέρματα που σημείωσε κάθε παίκτης ξεκινώντας από αυτόν που έχει πετύχει τα περισσότερα γκολ.

Δ. Να εμφανίζει τις ταξινομημένες λίστες.

3) Στο αγώνισμα των 100 μέτρων συμμετέχουν 10 αθλητές. Κάθε αθλητής έχει στη φανέλα του ένα νούμερο από το 101 μέχρι το 110. Αν φέτος έγιναν συνολικά 30 αγώνες να γραφεί πρόγραμμα Python το οποίο:

Α. Θα διαβάζει και θα τοποθετεί στη λίστα name το όνομα κάθε αθλητή.

Β. Θα διαβάζει και θα τοποθετεί στη λίστα num το νούμερο κάθε αθλητή με έλεγχο ώστε ο αριθμός να είναι μεταξύ 101 και 110 και να μην έχει δοθεί το ίδιο νούμερο σε άλλο αθλητή.

Γ. Θα διαβάζει και θα τοποθετεί στη λίστα vict το νούμερο του αθλητή που κέρδισε κάθε ένα από τους 30 αγώνες.

Δ. Θα εμφανίζει τον αριθμό των νικών κάθε αθλητή.

Ε. Θα εμφανίζει τα ονόματα των τριών πρώτων σε νίκες αθλητών.

ΑΣΚΗΣΗ 1

```
O=[]
T=[]
E=[]
for x in range(8):
   On=str(raw_input("Dose Onoma proiodos"))
   Tim=input('Dose timi')
   Et=str(raw_input('Dose eteria'))
   O.append(On)
   T.append(Tim)
   E.append(Et)
eteria=raw_input("Dose onoma eterias")

N=len(O)

for i in range(1,N,1):
   for j in range(N-1, i-1,-1):
      if T[j]<T[j-1]:
         T[j],T[j-1]=T[j-1],T[j]
         O[j],O[j-1]=O[j-1],O[j]
         E[j],E[j-1]=E[j-1],E[j]
for i in range(1,N,1):
   for j in range(N-1, i-1,-1):
      if T[j]==T[j-1] and O[j]<O[j-1]:
         T[j],T[j-1]=T[j-1],T[j]
         O[j],O[j-1]=O[j-1],O[j]
         E[j],E[j-1]=E[j-1],E[j]

for x in range(len(E)):
   if E[x]==eteria:
      print O[x], T[x]

if eteria not in E:
   print "Den yparxei eteria"
```

ΑΣΚΗΣΗ 2

```
name=[]
goal=[]
N=input("Dose arithmo pexton")
while N<0 or N>100:
   N=input("Dose SOSTO arithmo pexton")

for x in range(N):
   On=str(raw_input("Dose onoma pexti"))
   G=input("Dose arithmo goal")
   name.append(On)
   goal.append(G)

for i in range(1,N,1):
   for j in range(N-1, i-1,-1):
      if goal[j]>goal[j-1]:
         goal[j],goal[j-1]=goal[j-1],goal[j]
         name[j],name[j-1]=name[j-1],name[j]
print name
print goal
```

ΑΣΚΗΣΗ 3

```
name=[]
num=[]
vict=[]
nikes=[]
for x in range (10):
   On=str(raw_input('Dose onoma'))
   name.append(On)
   N=input('Dose numero athliti')
   while N<101 or N>110 or N in num:
      N=input('Dose SOSTO numero athliti')

   num.append(N)

for x in range(30):
   N1=input('Dose numero pou nikise')
   while N1<101 or N1>110 :
        N=input('Dose SOSTO numero athliti')
    vict.append(N1)
```

```
for x in range(10):
   c=0
   for y in vict:
      if y==num[x]:
         c=c+1
   print "O ", name[x], "exei ", c, "nikes"
   nikes.append(c)

for i in range(1,10,1):
   for j in range(10-1, i-1,-1):
      if nikes[j]>nikes[j-1]:
         nikes[j], nikes[j-1]=nikes[j-1],nikes[j]
         name[j], name[j-1]=name[j-1],name[j]
         num[j], num[j-1]=num[j-1],num[j]

print "1os athlitis =", name[0]
print "2os athlitis =", name[1]
print "3os athlitis =", name[2]
```

1) Σε έναν αγώνα φόρμουλα 1 κάθε οδηγός μπορεί να κάνει το πολύ 6 δοκιμαστικούς γύρους που χρονομετρούνται. Ο καλύτερος χρόνος από αυτούς είναι ο χρόνος με τον οποίο γίνεται η κατάταξη εκκίνησης του κανονικού αγώνα. (Ο ταχύτερος οδηγός ξεκινάει στη πρώτη θέση ο αμέσως επόμενος στη δεύτερη κοκ). Να γραφεί πρόγραμμα το οποίο:

Α. Θα διαβάζει και θα τοποθετεί στη λίστα driver το όνομα κάθε ενός από τους 24 οδηγούς που συμμετέχουν στον αγώνα.

Β. Για κάθε οδηγό θα διαβάζει επαναληπτικά το χρόνο του σε κάθε δοκιμαστικό γύρο. Η επανάληψη σταματά όταν ο χρήστης δώσει χρόνο 0 ή όταν ο οδηγός κάνει και τους έξι γύρους.

Γ. Θα υπολογίζει για κάθε αθλητή τον καλύτερο από τους παραπάνω χρόνους και θα τους τοποθετεί στην λίστα time.

Δ. Θα εμφανίζει τη σειρά εκκίνησης των αθλητών στον κανονικό αγώνα.

ΑΣΚΗΣΗ 1

```
driver=[]
time=[]

for x in range(24):
   On=str(raw_input("Dose onoma"))
   driver.append(On)
   c=0
   max1=0
   E=input("Dose epidosi")
   while E!=0 and c<=6:
      if E>max1:
         max1=E

      time.append(max1)
      c=c+1

N=len(time)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if time[j]>timr[j-1]:
         time[j],time[j-1]=time[j-1],time[j]
         driver[j],driver[j-1]=driver[j-1],driver[j]

print  "H katataksi  einai =", driver
```

1) Για τη διεκδίκηση πέντε (5) θέσεων υποτροφίας, εξετάστηκαν και βαθμολογήθηκαν πενήντα (50) υποψήφιοι σε τρία μαθήματα. Ο υπολογισμός του τελικού βαθμού κάθε υποψηφίου γίνεται ως εξής:

Αν ο βαθμός του σε κάποιο από τα δύο πρώτα μαθήματα είναι μικρότερος του 6, τότε ο τελικός βαθμός του είναι μηδέν (0). Διαφορετικά ο βαθμός του 1ου μαθήματος συμμετέχει στον υπολογισμό του τελικού βαθμού με συντελεστή 50%, ο βαθμός του 2ου μαθήματος με συντελεστή 30% και ο βαθμός του 3ου μαθήματος με συντελεστή 20%.

Να γραφεί πρόγραμμα Python το οποίο:

Α. Να διαβάζει και να τοποθετεί στη λίστα name το όνομα κάθε υποψηφίου.

Β. Να διαβάζει και να τοποθετήσει στις λίστες vath1, vath2, vath3 τον βαθμό κάθε υποψηφίου σε κάθε μάθημα, ελέγχοντας ότι είναι από 1 έως και 10.

Γ. Να υπολογίζει τον τελικό βαθμό κάθε υποψηφίου σύμφωνα με την παραπάνω διαδικασία και τον καταχωρίζει στη λίστα final.

Δ. Να εμφανίζει τα αποτελέσματα της διαδικασίας σύμφωνα με τα παρακάτω:

i. Επιλέγονται όλοι οι υποψήφιοι με βαθμό μεγαλύτερο από 8,5.

ii. Αν με την παραπάνω διαδικασία υπάρχουν ακόμα κενές θέσεις συμπληρώνονται από τους υποψηφίους με την καλύτερη βαθμολογία αρκεί αυτή να είναι τουλάχιστον 7. Στην περίπτωση που υπάρχουν υποψήφιοι που ισοβαθμούν με τον τελευταίο τότε για λόγους δικαιοσύνης επιλέγονται και αυτοί.

iii. Εμφανίζει τα ονόματα αυτών που επιλέχθηκαν.

Ε. Να εμφανίζει το ποσοστό των υποψηφίων που έχουν τελικό βαθμό μηδέν (0).

ΑΣΚΗΣΗ 1

```
name=[]
vath1=[]  ;  vath2=[] ; vath3=[]
final=[]

for x in range(50):
   On=str(raw_input('Dose Onoma'))
   name.append(On)
   V1=input('Dose Vathmo1')
   while V1<1 or V1>10:
      V1=input('Dose SOSTO  Vathmo1')
   V2=input('Dose Vathmo2')
   while V2<1 or V2>10:
      V2=input('Dose SOSTO  Vathmo2')
   V3=input('Dose Vathmo3')
   while V3<1 or V3>10:
      V3=input('Dose SOSTO  Vathmo3')
   vath1.append(V1)
   vath2.append(V2)
   vath3.append(V3)
z1=0
for x in range(50):
   if vath1[x]<6 or vath2[x]<6:
      final.append(0)
      z1=z1+1
   else:
      final.append(vath1[x]*50/100 + vath2[x]*30/100 + vath3[x]*20/100)

for i in range(1,50,1):
   for j in range(50-1,i-1,-1):
      if final[j]>final[j-1]:
         final[j],final[j-1]=final[j-1],final[j]
         vath1[j],vath1[j-1]=vath1[j-1],vath1[j]
         vath2[j],vath2[j-1]=vath2[j-1],vath2[j]
         vath3[j],vath3[j-1]=vath3[j-1],vath3[j]
         name[j],name[j-1]=name[j-1],name[j]

c=0
for x in range(50):
   if final[x]>8.5:
      c=c+1
      print name[x]
z=0
if c<5:
   D=5-c
   for x in range(c, 6, 1):
      if fina[x]>=7 and final[x]<=8.5:
         print name[x]
         z=z+1
for x in range(c+z,51,1):
   if final[x]==final[c+z-1]:
      print name[x]
print z1
```

1) Μια εταιρεία ασχολείται με εγκαταστάσεις ανεμογεννητριών, τις οποίες εγκαθιστά σε νησιά της Ελλάδας, της Ιταλίας, της Ισπανίας και της Γαλλίας. Κάθε νησί έχει την δυνατότητα αφενός να παράγει ηλεκτρική ενέργεια για να καλύπτει τις ανάγκες του νησιού, αφετέρου να πωλεί την πλεονάζουσα ενέργεια προς 0,60 €/kWh, εξασφαλίζοντας επιπλέον έσοδα.

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Δ1. Να διαβάζει για κάθε νησί το **όνομά** του και την **χώρα** και να δημιουργεί τον αντίστοιχο κωδικό του νησιού που αποτελείται από έναν αύξοντα αριθμό (η αρίθμηση ξεκινάει από το 1) και έχει κατάληξη τον κωδικό της χώρας (Ελλάδα GR, Ιταλία IT, Ισπανία ES, Γαλλία FR) π.χ. 12ES ή 13GR. Τα παραπάνω στοιχεία να τα καταχωρεί στις λίστες **ON** και **CODE** αντίστοιχα.

Δ2. Να διαβάζει για κάθε νησί το ποσό της ηλεκτρικής ενέργειας σε kWh που παρήγαγαν οι ανεμογεννήτριες κάθε νησιού, καθώς και το ποσό της ηλεκτρικής ενέργειας που κατανάλωσε κάθε νησί για **κάθε μήνα του έτους**.

Δ3. Να υπολογίζει την ετήσια παραγωγή και κατανάλωση ανά νησιού καθώς και τα ετήσια έσοδά του σε ευρώ (€) και να τα καταχωρεί στις λίστες **P** για την παραγωγή και **K** για την κατανάλωση και **E** για τα έσοδα αντίστοιχα. Θεωρήστε ότι για κάθε νησί η ετήσια παραγόμενη ηλεκτρική ενέργεια είναι μεγαλύτερη ή ίση της ενέργειας που έχει καταναλώσει.

Να διαβάζει δεδομένα μέχρι να δοθεί για όνομα νησιού η λέξη «TELOS» ή μέχρι να συμπληρωθούν συνολικά 25 νησιά.

Δ4. Να εμφανίζει το **όνομα** και τη **χώρα** του νησιού στο οποίο σημειώθηκε η μεγαλύτερη ετήσια παραγωγή ηλεκτρικού ρεύματος.

Δ5. Να καλεί κατάλληλο υποπρόγραμμα με τη βοήθεια του οποίου θα εμφανίζονται τα ονόματα και τα ετήσια έσοδα των νησιών κατά φθίνουσα σειρά ως προς τα ονόματα. Να κατασκευάσετε το υποπρόγραμμα που χρειάζεται για το σκοπό αυτό.

Δ6. Να δίνετε από το πληκτρολόγιο το όνομα ενός νησιού και στην συνέχεια να καλεί κατάλληλο υποπρόγραμμα **binarySearch** η οποία θα αναζητά το όνομα του νησιού και αν υπάρχει θα επιστρέφει την θέση του στην λίστα ON. Στην συνέχεια το νησί αυτό θα ελέγχεται αν είναι το νησί με τη μεγαλύτερη παραγωγή ηλεκτρικής ενέργειας και την μεγαλύτερη κατανάλωση. Αν δεν είναι να τυπώνεται κατάλληλο μήνυμα.

Αν δεν υπάρχει το νησί να εμφανίζεται κατάλληλο μήνυμα.

ΑΣΚΗΣΗ 1

```
def     binarysearch(ON, ONOMA) :
   first = 0
   last = len(ON)-1
   found = False
   while   found== False and  first <= last :

      mid = ( first + last ) // 2
      if    ON[ mid ] == ONOMA :
         found = True
      elif   ON[ mid ] < ONOMA :
         first = mid + 1
      else:
         last = mid-1

   if   found == True:
        return   mid
   else:
       return   -1


def BSO(ON, CODE, P, K, E):
   N=len(ON)
   for i in range(1,N,1):
      for j in range(N-1, i-1,-1):
         if ON[j] > ON[j-1]:
            E[j],E[j-1]= E[j-1], E[j]
            ON[j],ON[j-1]= ON[j-1], ON[j]
            CODE[j],CODE[j-1]= CODE[j-1], CODE[j]
            P[j],P[j-1]= P[j-1], P[j]
            K[j],K[j-1]= K[j-1], K[j]

   for x in range(len(ON)):
      print ON[x], E[x]

ON=[] ; CODE=[] ; P=[] ; K=[] ; E=[]

N=1
on=str(raw_input("Δώσε όνομα νησιού"))

while on!="TELOS" and N<=25:

   ON.append(on)

   xora=raw_input("Δώσε χώρα")
   if xora=="Ελλάδα":
      CODE.append(str(N)+"GR")
   if xora=="Ιταλία":
      CODE.append(str(N)+"IT")
   if xora=="Ισπανία":
      CODE.append(str(N)+"ES")
   if xora=="Γαλλία":
      CODE.append(str(N)+"FR")
```

```
   k=0
   p=0
   for x  in rnage(12):
      print "μηνας :", x+1
      pm=input("Δωσε παραγωγή")
      km=input("Δωσε καταναλωση")
      p=p+pm
      k=k+km

   P.append(p)
   K.append(k)

   e=p-k
   E.append(e*0.60)

   N=N+1
   on=str(raw_input("Δώσε όνομα νησιού"))

max1=0
for x in rnage(len(P)):
   if P[x]>max1:
      max1=P[x]
      Y=x

for x in rnage(len(CODE)):
   Z=CODE[x]
   if Z[len(CODE)-2 : ]=="GR":
      xora="Ελλαδα"
   if Z[len(CODE)-2 : ]=="IT":
      xora="Ιταλία"
   if Z[len(CODE)-2 : ]=="ES":
      xora="Ισπανία"
   if Z[len(CODE)-2 : ]=="FR":
      xora="Γαλλία"

print ON[Y], xora

BSO(ON, CODE, P, K, E)

ONOMA=raw_input("Δωσε ονομα νησιού")

V=binarysearch(ON, ONOMA)

if V==-1:
   print "Δεν υπάρχει το νησί"

else:
   max1=0
   max2=0
   for x in range(len(P)):
      if P[x]>max1:
         max1=P[x]
```

```
        Z1=x
    if K[x]>max2:
        max2=K[x]
        Z2=x
if Z1==Z2 and Z1==V :
    print "νησί με  τη μεγαλύτερη παραγωγή & κατανάλωση ηλεκτρικής ενέργειας"
else:
    print "οχι νησί με  μεγαλύτερη παραγωγή & κατανάλωση ηλεκτρικής ενέργειας"
```

1) Σε ένα σχολείο φοιτούν 120 μαθητές. Να γραφεί πρόγραμμα Python το οποίο:

Α. Να διαβάζει για κάθε μαθητή το ονοματεπώνυμο του, την τάξη του και τον τελικό βαθμό του και τα καταχωρεί στις αντίστοιχες λίστες NAME, TAKSI, και VATH ελέγχοντας την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:

- Οι τάξεις είναι Α ή Β ή Γ.
- Ο τελικός βαθμός είναι από 1 μέχρι και 20.

Β. Να εμφανίζει τα ονόματα των μαθητών της Β τάξης που έχουν τελικό βαθμό μεγαλύτερο του 18

Γ. Να υπολογίζει και να εμφανίζει το πλήθος των μαθητών κάθε τάξης,

Δ. Να υπολογίζει και εμφανίζει το μέσο όρο των τελικών βαθμών των μαθητών της Γ τάξης και

Ε. Να εμφανίζει ταξινομημένα κατά αλφαβητική σειρά τα ονοματεπώνυμα και τους αντίστοιχους τελικούς βαθμούς των μαθητών της Α τάξης.

ΑΣΚΗΣΗ 1

```
NAME=[ ] ; TAKSI=[ ] ; VATH=[ ]

for  x  in range(120):
   on=str(raw_input('Dose onoma'))
   T=str(raw_input('Dose taksi'))
   while T not in ["A","B","C"]:
      T=str(raw_input('Dose SOSTH taksi'))
   V=input('Dose vathmo')
   while V <0 or V > 20:
      V=input('Dose SOSTO vathmo')

   NAME.append(on)
   TAKSI.append(T)
   VATH.append(V)

for x  in range(120):
   if VATH[x]>18 and TAKSI[x]=="B":
      print NAME[x]

c1=0 ; c2=0;  c3=0
for x  in range(120):
   if TAKSI[x]=="A":
      c1=c1+1
   if TAKSI[x]=="B":
      c2=c2+1
   if TAKSI[x]=="C":
      c3=c3+1
print c1, c2, c3

S=0.0
for x  in range(120):
   if TAKSI[x]=="C":
      S=S+VATH[x]
print S/ c3

AT=[]
VA=[]
for x  in range(120):
   if TAKSI[x]=="A":
      AT.append(NAME[x])
      VA.append(VATH[x])

N=len(AT)
for  i in range(1, N,1):
   for j in range(N-1,i-1,-1):
      if AT[j]<AT[j-1]:
         AT[j],AT[j-1]=AT[j-1],AT[j]
         VA[j],VA[j-1]=VA[j-1],VA[j]

for x  in range(len(AT)):
   print AT[x], VA[x]
```

1) Ο υπεύθυνος καθηγητής ενός τμήματος θέλει να βγάλει ένα στατιστικό σχετικά με την επίδοση των μαθητών της τάξης του. Να γράψετε πρόγραμμα σε Python το οποίο :

Θα διαβάζει τα ονόματα των μαθητών και θα τα τοποθετεί σε μια ουρά ΟΝ μέχρι να πληκτρολογήσει για όνομα την λέξη 'ΤΕΛΟΣ'

Θα διαβάζει τους μέσους όρους των βαθμών που πήραν οι μαθητές (με έλεγχο ορθότητας τιμών ώστε οι βαθμοί να είναι μεταξύ 0 και 20) και θα τα τοποθετεί σε μια ουρά Μ

Θα ταξινομεί φθίνουσα με την μέθοδο ευθείας ανταλλαγής ως προς τον μέσο όρο

Θα τυπώνει τα ονόματα όσων έχουν τον μικρότερο βαθμό

Θα τυπώνει τον μέσο όρο του τμήματος

Θα τοποθετεί στην λίστα Α τα ονόματα όσων αρίστευσαν (18-20)

Θα τοποθετεί στην λίστα Β τα ονόματα όσων πήραν το χαρακτηρισμό λίαν καλώς (15-17)

Θα τοποθετεί στην λίστα C τα ονόματα όσων πήραν το χαρακτηρισμό Σχεδόν καλώς (10-14).

Θα τοποθετεί στην λίστα D τα ονόματα όσων πήραν μέσο όρο κάτω από 10.

Θα τυπώνει το ποσοστό επι της 100 όσων αρίστευσαν

Θα τυπώνει τον αριθμό των μαθητών που το όνομα τους ξεκινάει από 'Ε' ή 'Ν' και έχουν μέσο όρο κάτω από 10

Να δημιουργεί αντίγραφο της ουράς ΟΝ και Μ στις ουρές Ο1 και Μ1 αντίστοιχα.

Θα αδειάζει τις δύο ουρές ΟΝ και Μ

Θα αποθηκεύει τα ονόματα όσων πήραν το χαρακτηρισμό λίαν καλώς σε αρχείο με όνομα File1.txt.

θα προσθέτει στο αρχείο File1.txt τα ονόματα όσων πήραν το χαρακτηρισμό Σχεδόν καλώς.

θα δημιουργήσετε συνάρτηση η οποία θα δέχεται για είσοδο την λίστα C και θα επιστρέφει το όνομα του μαθητή με το μεγαλύτερο όνομα.

ΑΣΚΗΣΗ 1

```
def  MAXNAME(C):
   max1=0
   for x in C:
      if len(x)>max1:
         max1==len(x)
         z=x
   return z

ON=[]
M=[]

on=str(raw_input('Dose onoma'))
while  on!='TELOS':
   mo=input('Dose meso oro')
   while mo<0 or mo>20:
      mo=input('Dose SOSTO meso oro')
   ON.append(on)
   M.append(mo)
   on=str(raw_input('Dose onoma'))

N=len(M)
for  i in range(1, N,1):
   for j in range(N-1,i-1,-1):
      if M[j]>M[j-1]:
         M[j],M[j-1]=M[j-1],M[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
min1=M[0]
for x in range(1,len(M)):
   if M[x]<min1:
      min1=M[x]

for x in range(len(M)):
   if M[x]==min1:
      print ON[x]
S=0.0
for x in M:
   S=S+M
print "O mesos oros tou tmimatowQ",S/len(M)

A=[] ; B=[] ; C=[] ; D=[]
for x in range(len(M)):
   if M[x]>=18 and M[x]<=20:
      A.append(ON[x])
   if M[x]>=15 and M[x]<=17:
      B.append(ON[x])
   if M[x]>=10 and M[x]<=14:
      C.append(ON[x])
   if M[x]<10:
      D.append(ON[x])
```

```
print 'Pososto pou aristefsan:', 100*len(A)/len(M)

c=0
for x  in D:
   if x[0]=="E" or x[0]=="N":
      c=c+1
print  c

O1=ON[ : ]
M1=M[ : ]

while ON!=[]:
   ON.pop(0)
   M.pop(0)

f1=open("File1.txt","w")
for  x  in C:
   f1.write(x + "\n")
f1.close()


print "Megalitero onoms:", MAXNAME(C)
```

1) Το Κράτος ανακοίνωσε τον παρακάτω πίνακα που αφορά τα Τέλη κυκλοφορίας που πρέπει να πληρώσουν οι πολίτες ανάλογα με τον κυβισμό και την παλαιότητα του αυτοκινήτου.

	Παλαιότητα	
Κυβισμός αυτοκινήτου	**1-10 έτη**	**Πάνω από 10 έτη**
Κάτω από 1000 κυβικά εκατοστά	90 €	120 €
Από 1000 μέχρι και 1999	130 €	240 €
Πάνω από 1999	190 €	300 €

Μία εταιρεία διαθέτει 20 αυτοκίνητα. Να γράψετε πρόγραμμα σε Python το οποίο:

α) Για κάθε αυτοκίνητο να διαβάζει τον αριθμό κυκλοφορίας και να τον αποθηκεύει σε μία λίστα AR

β) να διαβάζει την παλαιότητα και τα κυβικά εκατοστά του αυτοκινήτου και να υπολογίζει το ποσό που πρέπει να πληρώσουμε για το συγκεκριμένο αυτοκίνητο.

γ) Στη συνέχεια να τοποθετεί αυτό το ποσό σε μία λίστα POSO

δ) να υπολογίζει και να εμφανίζει το συνολικό ποσό που πρέπει να πληρώσει η εταιρεία

ε) να υπολογίζει και να εμφανίζει το μικρότερο ποσό που πρέπει να πληρώσει για ένα αυτοκίνητο.

στ) να υπολογίζει και να εμφανίζει όλους τους αριθμούς κυκλοφορίας των αυτοκινήτων για τους οποίους θα πληρώσουμε αυτό το μικρότερο ποσό.

ζ) να υπολογίζει και να εμφανίζει όσους αριθμούς κυκλοφορίας ξεκινούν με “KZN” και να εμφανίζει αυτούς και το ποσό που πρέπει να πληρώσει για το κάθε αυτοκίνητο (Θεωρούμε ότι όλοι οι αριθμοί κυκλοφορίας ξεκινούν με 3 κεφαλαία ελληνικά γράμματα).

2) Σε ένα διαγωνισμό ενός video game υπάρχει συμμετοχή αγοριών και κοριτσιών.

Να γράψετε πρόγραμμα σε python το οποίο:

α) να διαβάζει το φύλο (“Α” ή “Κ”) του ή της διαγωνιζόμενης και τη βαθμολογία και να τα καταχωρίζει στις λίστες ΒΑ και FY αντίστοιχα

β) η διαδικασία να επαναλαμβάνεται μέχρι να δοθεί για φύλο η λέξη “TELOS”

γ) να διαχωρίζει τη λίστα ΒΑ σε δύο λίστες μία ΒΑΑ για τα αγόρια και μία ΒΑΚ για τα κορίτσια.

δ) Με τη βοήθεια της bubblesort να υπολογίζει και να εμφανίζει το μέγιστο και τον ελάχιστο βαθμό των αγοριών και το μέγιστο και τον ελάχιστο βαθμό των κοριτσιών.

3) Να γράψετε πρόγραμμα σε python το οποίο:

α) να έχει συνάρτηση countS που να δέχεται σαν είσοδο μία λέξη και να επιστρέφει το πλήθος των συμφώνων (Στα Ελληνικά μικρά ή κεφαλαία) που περιέχει

β) να έχει συνάρτηση bubbleSort η οποία να δέχεται μία λίστα και να την ταξινομεί σε αύξουσα σειρά.

γ) να διαβάζει από το πληκτρολόγιο λέξεις μέχρι να μας δοθεί η λέξη " TELOS" και να τις καταχωρεί στη λίστα WORDS.

δ) με τη βοήθεια της παραπάνω συνάρτησης countS να εμφανίζει τη λέξη με το μέγιστο πλήθος συμφώνων

ε) με τη βοήθεια της bubbleSort να ταξινομεί τη λίστα WORDS σε αύξουσα σειρά (αλφαβητική)

στ) να εγγράφει όλες τις λέξεις με αύξουσα (αλφαβητική) σειρά, μία σε κάθε γραμμή σε ένα νέο αρχείο "lexeis.txt" έχοντας από μπροστά και τον αριθμό της κάθε σειράς (ξεκινώντας από 1).

4) Σε ένα διαγωνισμό του δημοσίου οι διαγωνιζόμενοι διαγωνίζονται σε δύο μαθήματα: Τη γλώσσα και τα Μαθηματικά. Να γράψετε πρόγραμμα σε Python το οποίο για κάθε διαγωνιζόμενο:

α) να διαβάζει το Επώνυμο και το βαθμό στο μάθημα της Γλώσσας και των Μαθηματικών και να τους καταχωρίζει σε 3 λίστες EP, GL και MA αντίστοιχα

β) Η διαδικασία να επαναλαμβάνεται μέχρι να μας δοθεί για επώνυμο η λέξη "TELOS"

γ) να υπολογίζει και να εμφανίζει πόσο έγραψαν κατά μέσο όρο οι μαθητές σε κάθε μάθημα.

δ) να υπολογίζει και να εμφανίζει το ΜΟ από τα δύο μαθήματα για κάθε εξεταζόμενο και να τον καταχωρεί σε μία λίστα MESOS.

ε) Να εμφανίζει πόσοι διαγωνιζόμενοι έχουν πάνω από 10 μέσο όρο στα δύο μαθήματα.

στ) Να υπολογίζει πόσοι διαγωνιζόμενοι έγραψαν πιο πολύ στα μαθηματικά από ότι στη γλώσσα

ΑΣΚΗΣΗ 1

```
AR=[]
POSO=[]
for i in range(20):

   ar=raw_input("Δώσε τον αριθμό κυκλοφορίας")
   AR.append(ar)

   pal=input("Δώσε την παλαιότητα του αυτοκινήτου")
   ke=input("Δώσε τα κυβικά εκατοστά ")
   if pal>=1 and pal<=10:
      if ke<1000:
         poso=90
      if ke>=1000 and ke<=1999:
         poso=130
      if ke>1999:
         poso=190
   elif pal>10:
      if ke<1000:
         poso=120
      if ke>=1000 and ke<=1999:
         poso=240
      if ke>1999:
         poso=300

   POSO.append(poso)

SUM=0.0
for i in range(len(POSO)):
   SUM=SUM+POSO[i]
print "Το συνολικό ποσό που πρέπει να πληρώσει η εταιρεία είναι ",SUM

MIN=POSO[0]
for i in range(1,len(POSO)):
   if POSO[i]<MIN:
      MIN=POSO[i]
print "Το μικρότερο ποσό που πρέπει να πληρώσει η εταιρεία είναι ", MIN

print "Τα οχήματα που θα πληρώσουν ", MIN, "€ είναι :"
for i in range(len(POSO)):
   if POSO[i]==MIN:
      print AR[i]

for i in range(len(AR)):
   arithmos=AR[i]
   if arithmos[ : 3]=="KZN":
      print arithmos, POSO[i]
```

ΑΣΚΗΣΗ 2

```
BA=[]
FY=[]
f=raw_input("Δώσε το φύλο")

while f!="TELOS":
   b=float(input("Δώσε το βαθμό"))
   FY.append(f)
   BA.append(b)
   f=raw_input("Δώσε το φύλο")

BAA=[]
BAK=[]
for i in range(len(BA)):
   if FY[i]=="K":
      BAK.append(BA[i])
   else:
      BAA.append(BA[i])
print "Οι βαθμοί των αγοριών ",BAA
print "Οι βαθμοί των κοριτσιών ",BAK


N=len(BAA)
for i in range(N-1):
   for j in range(N-1,i,-1):
      if BAA[j]<BAA[j-1]:
         BAA[j],BAA[j-1]=BAA[j-1],BAA[j]

print " Ο μέγιστος βαθμός των αγοριών είναι ο ",BAA[-1]," και ο ελάχιστος ο ",BAA[0]

N=len(BAK)
for i in range(N-1):
   for j in range(N-1,i,-1):
      if BAK[j]<BAK[j-1]:
         BAK[j],BAK[j-1]=BAK[j-1],BAK[j]
print " Ο μέγιστος βαθμός των κοριτσιών είναι ο ",BAK[-1]," και ο ελάχιστος ο ",BAK[0]
```

ΑΣΚΗΣΗ 3

```
def countS(word):
   symfona="ΒΓΔΖΘΚΛΜΝΞΠΡΣΤΦΧΨβγδζθκλμνξπρστφχψ"
   count=0
   for letter in word:
      if letter in symfona:
         count=count+1
   return count


def bubbleSort(A):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]

WORDS=[]
word=str(raw_input("Δώσε τη λέξη"))
while word!="TELOS":
   WORDS.append(word)
   word=str(raw_input("Δώσε τη λέξη"))

MAX=countS(WORDS[0]) # Θα μπορούσε να είναι και MAX=-1
MAXLEXI=WORDS[0] #Θα μπορούσε να είναι και " "
for i in range(len(WORDS)):
   arithmos_symfonwn=countS(WORDS[i])
   if arithmos_symfonwn>MAX:
      MAX=arithmos_symfonwn
      MAXLEXI=WORDS[i]
print "Η λέξη με τα περισσότερα σύμφωνα είναι ",MAXLEXI," και έχει ",MAX," σύμφωνα"

bubbleSort(WORDS)

for i in range(len(WORDS)):
   print WORDS[i]

f=open("lexeis.txt","w")

for i in range(len(WORDS)):
   f.write(str(i+1)+"."+WORDS[i]+"\n")

f.close()
```

ΑΣΚΗΣΗ 4

```
EP=[]
GL=[]
MA=[]
ep=raw_input("Δώσε το επώνυμο")

while ep!="TELOS":
   glo=float(input("Δώσε το βαθμό στη Γλώσσα"))
   math=float(input("Δώσε το βαθμό στα Μαθηματικά"))
   EP.append(ep)
   GL.append(glo)
   MA.append(math)

   ep=raw_input("Δώσε το επώνυμο")

SUMGL=0
SUMMA=0
for i in range(len(EP)):
   SUMGL=SUMGL+GL[i]
   SUMMA=SUMMA+MA[i]

MOGL=SUMGL/len(GL)
MOMA=SUMMA/len(MA)
print "Ο μέσος όρος στη γλώσσα είναι ",MOGL
print "Ο μέσος όρος στα μαθηματικά είναι ",MOMA

MESOS=[]
for i in range(len(EP)):
   mesosoros=(GL[i]+MA[i])/2.0
   MESOS.append( mesosoros)
   print "Ο/Η διαγωνιζόμενος/νη ", EP[i]," έχει μέσο όρο από τα δύο μαθήματα ",mesosoros

m1=0
for i in range(len(EP)):
   if MESOS[i]>10:
      m1=m1+1
print "Πάνω από 10 μέσο όρο έχουν ", m1," διαγωνιζόμενοι/νες"

m2=0
for i in range(len(EP)):
   if MA[i]>GL[i]:
      m2=m2+1
print 'Έγραψαν καλύτερα στα Μαθηματικά απ' ότι στη Γλώσσα ", m1," διαγωνιζόμενοι/νες"
```

1) Να γράψετε πρόγραμμα σε python το οποίο:

α) να διαβάζει το όνομα μιας πόλης (ή χωριού), το νομό στον οποίο ανήκει (με κεφαλαία Ελληνικά γράμματα και τα δύο) και τον πληθυσμό της και να τα καταχωρίζει στις λίστες ON, NOM, PL αντίστοιχα.

β) να επαναλαμβάνει τη διαδικασία μέχρι να δοθεί για όνομα πόλης η λέξη “ΤΕΛΟΣ”

γ) να εμφανίζει το όνομα και το νομό για όσες πόλεις (χωριά) έχουν πληθυσμό πάνω από 4.000 κατοίκους

δ) να υπολογίζει και να εμφανίζει πόσο είναι το σύνολο των κατοίκων από όλες τις πόλεις (χωριά)

ε) με τη βοήθεια της συνάρτησης bubbleSort να ταξινομεί τις πόλεις με φθίνουσα ταξινόμηση και να εμφανίζει τις 3 πόλεις με το μεγαλύτερο πληθυσμό γράφοντας για κάθε πόλη: το όνομά της, το νομό και τον πληθυσμό της

στ) να δημιουργεί ένα αρχείο “thess.txt” μέσα στο οποίο να εγγράφει για κάθε πόλη (χωριό) που βρίσκεται στο νομό ΘΕΣΣΑΛΟΝΙΚΗΣ , σε κάθε μία γραμμή το όνομά του και τον πληθυσμό του.

ζ) Να εμφανίζει το όνομα και τον πληθυσμό των πόλεων που το όνομά τους ξεκινά με το κείμενο “ΝΕΑ” π.χ. ΝΕΑ ΑΓΧΙΑΛΟΣ 1500

2) Φορτώνουμε ένα πλοίο με 300 κοντέινερ. Αυτά μπορεί να είναι τριών ειδών: το τύπου Α, το τύπου Β και το τύπου C (να γίνεται έλεγχος ορθότητας). Το μέγιστο μεικτό βάρος του κάθε κοντέινερ δεν πρέπει να ξεπερνάει τα 32500 Kgr. Να γράψετε πρόγραμμα σε python το οποίο:

α) Να διαβάζει τον τύπο του κοντέινερ και το μεικτό βάρος του σε Kgr.

β) Να κάνει έλεγχο ορθότητας στο βάρος του κάθε κοντέινερ ώστε τα κιλά που καταχωρίζονται να είναι πάνω από 0 και κάτω από 32500 Kgr

γ) Να υπολογίζει και να εμφανίζει το σύνολο των κιλών που φορτώθηκαν στο πλοίο.

δ) Να υπολογίζει πόσα κιλά ήταν βαρύ κατά μέσο όρο το κάθε κοντέινερ

ε) Να υπολογίζει και να εμφανίζει το κοντέινερ με το μεγαλύτερο βάρος και τον τύπο που είχε (θεωρώντας ότι είναι μοναδικό).

στ) Να υπολογίζει και να εμφανίζει το κοντέινερ με το μικρότερο βάρος και τον τύπο που είχε (θεωρώντας ότι είναι μοναδικό).

ζ) Να υπολογίζει και να εμφανίζει πόσα κοντέινερ φορτώθηκαν από κάθε τύπο.

η) Να υπολογίζει και να εμφανίζει το συνολικό βάρος που αντιστοιχεί στον κάθε τύπο. Δηλαδή πόσα κιλά ήταν όλα τα κοντέινερ του τύπου Α μαζί, πόσα ήταν του Β και πόσα του C.

3) Μία εταιρία παραγωγής και εμπορίας φέτας διαθέτει ένα συγκεκριμένο τύπο φορτηγού που μπορεί να χωρέσει μέχρι και 1000 κιλά βάρος. Θεωρούμε ότι η εταιρία φορτώνει τα φορτηγάκια της με δύο είδη δοχείων. Τα δοχεία τύπου Α που είναι ολόκληρα και ζυγίζουν 17 κιλά και τα δοχεία τύπου Β που είναι μισά και ζυγίζουν 8,5 κιλά.

Να γράψετε πρόγραμμα σε python το οποίο:

α) Να διαβάζει τον τύπο του δοχείου που φορτώνουμε. Η διαδικασία να επαναλαμβάνεται μέχρι το συνολικό βάρος που φορτώθηκε στο φορτηγάκι να ξεπεράσει τα 1000 Kgr. (Δεν χρειάζεται έλεγχος ορθότητας)

β) Να υπολογίζει και να εμφανίζει πόσα δοχεία τύπου Α και πόσα τύπου Β φορτώθηκαν.

γ) Να υπολογίζει και να εμφανίζει πόσα κιλά μπήκαν συνολικά σε δοχεία τύπου Α και πόσα σε δοχεία τύπου Β.

δ) Να υπολογίζει τα συνολικά έσοδα από την πώληση των δοχείων αν κάθε δοχείο τύπου Α πωλείται προς 100 € και κάθε δοχείο τύπου Β προς 50 €.

ΑΣΚΗΣΗ 1

```
def   bubbleSort(A,B,C):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if A[j]>A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
            C[j],C[j-1]=C[j-1],C[j]
ON=[]
NOM=[]
PL=[]
on=raw_input("ΠΟΛΗ ή ΧΩΡΙΟ: ")

while on!="ΤΕΛΟΣ":
   nomos=raw_input("ΝΟΜΟΣ: ")
   pl=int(input("ΠΛΗΘΥΣΜΟΣ: "))
   ON.append(on)
   NOM.append(nomos)
   PL.append(pl)
   on=raw_input("ΠΟΛΗ ή ΧΩΡΙΟ: ")

print "Χωριά ή Πόλεις με πληθυσμό πάνω από 4000 κατοίκους"
for i in range(len(ON)):
   if PL[i]>4000:
      print ON[i],NOM[i]

SUM=0
for i in range(len(ON)):
   SUM=SUM+PL[i]
print "Ο συνολικός πληθυσμός είναι ",SUM

bubbleSort(PL,ON,NOM)
print "Οι 3 πόλεις με το μεγαλύτερο πληθυσμό είναι :"
print "ΟΝΟΜΑ: ",ON[0]," ΝΟΜΟΣ: ", NOM[0]," ΠΛΗΘΥΣΜΟΣ:", PL[0]
print "ΟΝΟΜΑ: ",ON[1]," ΝΟΜΟΣ: ", NOM[1]," ΠΛΗΘΥΣΜΟΣ:", PL[1]
print "ΟΝΟΜΑ: ",ON[2]," ΝΟΜΟΣ: ", NOM[2]," ΠΛΗΘΥΣΜΟΣ:", PL[2]

f=open("thess.txt","w")
for i in range(len(ON)):
   if NOM[i]=="ΘΕΣΣΑΛΟΝΙΚΗΣ":
      f.write(ON[i]+" "+str(PL[i])+"\n")
f.close()

for i in range(len(ON)):
   ONOMA=ON[i]
   if ONOMA[  :3]=="ΝΕΑ":
      print ON[i], PL[i]
```

ΑΣΚΗΣΗ 2 - Χωρίς Λίστες

```
SUM=0
MAX=0
MAXTYPOS=" "
MIN=32501
MINTYPOS=" "
mA=mB=mC=0
SUMA=SUMB=SUMC=0
for i in range(300):
   t=raw_input("Δώσε τον τύπο του κοντέινερ (A ή B ή C):")
   while t not in ['A','B', 'C']:
        t=raw_input("Δώσε τον σωστό τύπο του κοντέινερ (A ή B ή C):")
   mb=float(input("Δώσε το μεικτό βάρος του κοντέινερ:"))
   while mb<=0 or mb>=32500:
        mb=float(input("Δώσε το σωστό μεικτό βάρος του κοντέινερ :"))
   SUM=SUM+mb

   if mb>MAX:
      MAX=mb
      MAXTYPOS=t
   if mb<MIN:
     MIN=mb
     MINTYPOS=t

   if t=="A":
      mA=mA+1
      SUMA=SUMA+mb
   if t=="B":
      mB=mB+1
      SUMB=SUMB+mb
   if t=="C":
      mC=mC+1
      SUMC=SUMC+mb

print "Το σύνολο των κιλών που φορτώθηκαν στο πλοίο είναι ", SUM

MO=SUM/300.0
print " Το κάθε κοντέινερ είναι κατά μέσο όρο ", MO," κιλά"

print "Το φορτίο με το μέγιστο βάρος είναι ", MAX,"κιλά και είναι τύπου",MAXTYPOS
print "Το φορτίο με το ελάχιστο βάρος είναι ", MIN,"κιλά και είναι τύπου",MINTYPOS

print "Έχω ", mA," κοντέινερ τύπου A"
print "Έχω ", mB," κοντέινερ τύπου B"
print "Έχω ", mC," κοντέινερ τύπου C"

print " Τα κοντέινερ τύπου A έχουν συνολικό βάρος", SUMA
print " Τα κοντέινερ τύπου B έχουν συνολικό βάρος", SUMB
print " Τα κοντέινερ τύπου C έχουν συνολικό βάρος", SUMC
```

ΑΣΚΗΣΗ 3 - Χωρίς Λίστες

```
SUM=0.0
mA=mB=0
while SUM<=1000:
   typos=raw_input("Δώσε τον τύπο του δοχείου (A ή B)")
   if typos=="A":
      SUM=SUM+17
      mA+=1       #Είναι το ίδιο με το mA=mA+1
   else:
      SUM=SUM+8.5
      mB+=1      #Είναι το ίδιο με το mB=mB+1

print "Φορτώθηκαν ", mA,' δοχεία τύπου A και ', mB,' δοχεία τύπου B'
# Για να βρω τα κιλά που μπήκαν σε δοχεία τύπου A (SUMA) πολλαπλασιάζω τα δοχεία τύπου A επί 17 (γ)
SUMA=mA*17
# Για να βρω τα κιλά που μπήκαν σε δοχεία τύπου B (SUMB) πολλαπλασιάζω τα δοχεία τύπου B επί 8.5 (γ)
SUMB=mB*8.5

print "Τα κιλά που μπήκαν σε δοχεία τύπου A είναι ", SUMA
print "Τα κιλά που μπήκαν σε δοχεία τύπου B είναι ", SUMB
#Υπολογίζω τα συνολικά έσοδα se (δ)
se=mA*100+mB*50

print "Τα συνολικά έσοδα είναι ", se," €"
```

1) Σε λίστα 1000 θέσεων διαβάζονται με τη βοήθεια αλγορίθμου οι ονομασίες των προϊόντων ενός πολυκαταστήματος. Η θέση στην λίστα κάθε προϊόντος δείχνει και τον κωδικό του. Δηλ. το προϊόν με κωδικό i αποθηκεύεται στην i-οστη θέση της λίστας. Σε δεύτερη λίστα διαβάζεται η τιμή κάθε προϊόντος και σε τρίτη λίστα ο αριθμός τεμαχίων που διαθέτει το κατάστημα ως απόθεμα.

Αφού διαβαστούν τα δεδομένα ο αλγόριθμος θα διευκολύνει τον υπεύθυνο καταστήματος στα ακόλουθα:

α) Ο αλγόριθμος θα διαβάζει τον κωδικό ενός προϊόντος και θα εμφανίζει τα υπόλοιπα τρία στοιχεία του.

β) Ο αλγόριθμος θα εμφανίζει τα στοιχεία των προϊόντων χωρίς απόθεμα.

γ) Ο αλγόριθμος θα εμφανίζει τα στοιχεία των προϊόντων με το μεγαλύτερο απόθεμα.

δ) Ο αλγόριθμος θα διαβάζει την ονομασία ενός προϊόντος και αν υπάρχει στην αποθήκη θα εμφανίζει το απόθεμά του. Διαφορετικά θα ενημερώνει με σχετικό μήνυμα για την μη ύπαρξη του προϊόντος.

ε) Να εμφανίζει τα στοιχεία των λιστών σε φθίνουσα σειρά αποθέματος. Σε περίπτωση που κάποια προϊόντα έχουν ίδιο απόθεμα, να εμφανίζονται πρώτα τα προϊόντα αλφαβητικά ταξινομημένα.

ζ) Στη συνέχεια να γράψετε τη **συνάρτηση binarySearch(lista, key)** η οποία να επιστρέφει τη θέση του key μέσα στο lista. Με την βοήθεια της να ξαναλύσετε την δ

ΑΣΚΗΣΗ 1

```
def    binarysearch(ON, ONOMA) :
   first = 0
   last = len(ON)-1
   found = False
   while   found== False and  first <= last :

      mid = ( first + last ) // 2
      if    ON[ mid ] == ONOMA :
         found = True
      elif   ON[ mid ] < ONOMA :
         first = mid + 1
      else:
         last = mid-1

   if   found == True:
        return   mid
   else:
       return   -1

ON=["p1", "p2","p3"]
TM=[100, 110, 50]
T=[2, 5, 0]

K= input("Dose kodiko")
print ON[K], TN[K], T[K]

for x in range(1000):
   if T[x]==0:
      print ON[x], TM[x], T[x]

max1=T[0]
for x in range(1,1000,1):
   if T[x]>max1:
      max1=T[x]

for x in range(1000):
   if T[x]== max1:
      print ON[x], TNM[x], T[x]

Y=True
On=raw_input('Dose proion')
for x in range(1000):
   if ON[x]==On:
      print T[x]
      Y=False

if Y==True:
   print "Den yparxei"
```

```
N=1000

for i in range(1,N,1):

   for j in range(N-1,i-1,-1):

      if   T[j]>T[j-1]:
         T[j],T[j-1]=T[j-1],T[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         TM[j],TM[j-1]=TM[j-1],TM[j]

      if   T[j]==T[j-1]   and   ON[j]>ON[j-1]:
         T[j],T[j-1]=T[j-1],T[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         TM[j],TM[j-1]=TM[j-1],TM[j]

ONOMA=raw_input('Dose proion')


if  binarysearch(ON, ONOMA) ==-1:
   print 'Den yparxei'
else:
   print T[binarysearch(ON, ONOMA)]
```

1) Κατά την απογραφή του έτους 2001 σε ένα χωριό απογράφηκαν 1800 άτομα.

Να γραφεί αλγόριθμος με τον οποίο:

Α. Θα αποθηκεύεται το έτος γέννησης όλων των ατόμων σε μια λίστα και το ονοματεπώνυμο σε μια άλλη λίστα με αντιστοιχία θέσεων.

Β. Θα υπολογίζεται και θα εμφανίζεται το πλήθος των ατόμων κατά ηλικία συνοδευόμενο από τη φράση:

0 έως και 25 χρόνων: "ΝΕΟΙ"

26 έως και 50 χρόνων: "ΜΕΣΗΛΙΚΕΣ"

άνω των 50 χρόνων: "ΓΕΡΟΝΤΕΣ"

Γ. Θα εμφανίζονται οι ηλικίες των 4 μεγαλύτερων ατόμων.

Δ. Θα εμφανίζονται τα ονόματα των κατοίκων που έχουν τις 4 μεγαλύτερες ηλικίες.

2) Μια εταιρεία κρατά σε λίστα τα ονόματα 100 προμηθευτών καθώς και τα χρήματα που χρωστάει στον καθένα.

Να φτιάξετε πρόγραμμα με το οποίο να καταχωρούνται τα παραπάνω στοιχεία σε δύο λίστες Α και Β.

Να εμφανίζει το ή τα ονόματα των προμηθευτών στους οποίους η εταιρεία χρωστά τα περισσότερα χρήματα.

Να μπορεί να δέχεται ως είσοδο το όνομα ενός προμηθευτή και να εμφανίζει το ποσό που του χρωστάει. Αν το όνομα δεν υπάρχει να εμφανίζει μήνυμα ότι ο συγκεκριμένος προμηθευτής δεν υπάρχει.

3) Ένα κατάστημα παιχνιδιών διαθέτει 10000 διαφορετικά προϊόντα προς πώληση. Να φτιάξετε πρόγραμμα το οποίο διαβάζει τους κωδικούς, το όνομα του παιχνιδιού και το όνομα του προμηθευτή του αντίστοιχου παιχνιδιού και να τα τοποθετεί σε 3 λίστες Κ, ΟΝ και Ρ.

Στη συνέχεια να διαβάζει από το πληκτρολόγιο το όνομα ενός προμηθευτή και μετά από αναζήτηση να εμφανίζει όλα τα διαθέσιμα παιχνίδια (κωδικό και όνομα παιχνιδιού) που αντιστοιχούν σε αυτόν. Σε περίπτωση που ο προμηθευτής δεν βρεθεί να εμφανίζει κατάλληλο μήνυμα στην οθόνη.

4) Να γράψετε πρόγραμμα το οποίο να διαβάζει μια λίστα 100 πραγματικούς αριθμούς και να εμφανίζει τους 10 μικρότερους.

5) Να εισάγετε σε λίστες Α και Β τα ονόματα 100 μαθητών και τους βαθμούς τους σε ένα μάθημα.

Στη συνέχεια να δημιουργήσετε τις λίστες Α1 και Β1 100 θέσεων και γεμίστε με παύλες ' - ' την Α1 και με -1 την Β1.

Το πρόγραμμα που θα φτιάξετε να ελέγχει ποιοι μαθητές έχουν βαθμό μεγαλύτερο από 17,5 και να τοποθετεί τα ονόματα και τους βαθμούς τους στις πρώτες θέσεις των λιστών Α1 και Β1 αντίστοιχα.

Τέλος να γίνει φθίνουσα ταξινόμηση ως προς την λίστα Β1 και να εμφανίσετε τα ονόματα και τους βαθμούς των μαθητών με βαθμό πάνω απο 17,5.

6) Να δίνετε από το πληκτρολόγιο τον αριθμό των προϊόντων ενός Super Market. Για κάθε προϊόν να πληκτρολογείτε το όνομα του, την τιμή του και την εταιρεία του προϊόντος.

Δημιουργήστε 3 λίστες ΟΝ, Τ, Ε για την αποθήκευση στοιχείων των προϊόντων. Η λίστα ΟΝ για τα ονόματα, η Τ για τις τιμές και η Ε για την εταιρεία του κάθε προϊόντος.

Να διαβάζεται κατά την εκτέλεση του προγράμματος το όνομα μιας εταιρείας.

Στη συνέχεια να βρίσκονται και να εμφανίζονται με αύξουσα σειρά ως προς την τιμή τους όλα τα στοιχεία για τα προϊόντα της συγκεκριμένης εταιρείας. Σε περίπτωση που κάποια προϊόντα έχουν την ίδια αξία τότε να εμφανίζεται πρώτο το προϊόν με το μικρότερο αλφαβητικά όνομα. Τέλος αν δεν βρεθεί η εταιρεία που εισάγατε να εμφανίζεται κατάλληλο μήνυμα.

7) Να δίνετε από το πληκτρολόγιο τον αριθμό των καταστημάτων της Βόρειας Ελλάδας και τον αριθμό των καταστημάτων της Νότιας Ελλάδας μιας εταιρείας. Ο αριθμός των καταστημάτων κάθε περιοχής δεν μπορεί να ξεπερνά τα 45 (Να γίνεται έλεγχος ορθότητας)

Για την περιοχή της Βόρειας Ελλάδας να αποθηκεύεται σε λίστα Β το όνομα κάθε καταστήματος αντίστοιχα για την περιοχή της Νότιας Ελλάδας να αποθηκεύεται σε λίστα Ν το όνομα κάθε καταστήματος.

Να δημιουργήσετε και να χρησιμοποιήσετε την συνάρτηση bubblesort και να ταξινομήσετε με αυτή κατά φθίνουσα σειρά τις λίστες Β και Ν.

Τέλος να συνενώσετε τις λίστες Β και Ν σε μια νέα ταξινομημένη λίστα Α.

8) Η στατιστική υπηρεσία της ευρωπαϊκής ένωσης για μια μελέτη σχετικά με τον πληθυσμό στα ευρωπαϊκά κράτη διατηρεί λίστα Α με τα ονόματα των 27 κρατών – μελών και λίστες Ρ_2019 και Ρ_2020 με τους πληθυσμούς των κρατών αυτών για τα έτη 2019 και 2020 αντίστοιχα.

Να αναπτύξετε αλγόριθμο που με δεδομένους της παραπάνω λίστες:

Α. Να εμφανίζει την επί τοις εκατό αύξηση του πληθυσμού στην ευρωπαϊκή ένωση από το 2019 στο 2020.

Β. Να εμφανίζει για κάθε κράτος την επί τοις εκατό αύξηση του πληθυσμού από το 2019 στο 2020. Ποιο κράτος είχε τη μεγαλύτερη επί τοις εκατό αύξηση.

Γ. Υπάρχει κράτος που είχε τους περισσότερους κατοίκους τα δυο αυτά έτη; Αν ναι, τότε ποιο είναι αυτό;

9) Μια εταιρεία κινητής τηλεφωνίας κάνει δειγματοληπτικό έλεγχο πελατών. Να γραφεί πρόγραμμα, που θα διαβάζει τον αριθμό λεπτών που μίλησαν Ν πελάτες της εταιρείας μέσα σε ένα έτος και να τοποθετεί τις τιμές στην λίστα Α και το όνομα του κάθε πελάτη και να το τοποθετεί στην λίστα Β.

Το πλήθος Ν των πελατών θα ζητείται από τον χρήστη.

Να δημιουργηθεί δεύτερη λίστα C που θα περιέχει τη χρέωση σε Ευρώ για κάθε πελάτη. Η χρέωση είναι κλιμακωτή ως εξής:

▪ 0-15 λεπτά 0.45λεπτά του Ευρώ ανά λεπτό,

▪ στα υπόλοιπα λεπτά η χρέωση είναι 0.30 ανά λεπτό.

α) Να υπολογιστεί και εμφανιστεί η συνολική χρέωση για το έτος ανά πελάτη.

β) Να υπολογιστεί και εμφανιστεί το συνολικό καθαρό κέρδος της εταιρείας. Η εταιρεία έχει καθαρό κέρδος ίσο με 15% του συνολικού ποσού που εισέπραξε στη διάρκεια του έτους.

γ) Να τυπωθούν τα ονόματα των 10 πελατών με την μικρότερη χρέωση

10) Ένας σκληρός δίσκος έχει χωρητικότητα 500 MB για αποθήκευση αρχείων. Ο κάτοχός του τον γεμίζει με αρχεία. Θεωρώντας ότι το αποθηκευτικό μέσο είναι αρχικά άδειο, να γράψετε, σε Python, πρόγραμμα που θα διαβάζει το μέγεθος κάθε αρχείου σε MB, μέχρι το συνολικό μέγεθος να ξεπεράσει τη χωρητικότητά αυτή. Στη συνέχεια, θα εμφανίζει το συνολικό πλήθος των αρχείων που έχουν αποθηκευθεί στο δίσκο.

ΑΣΚΗΣΗ 1

```
A=[]
B=[]
H1=[]

for x in range(1800):
   On=str(raw_input('Dose Onoma'))
   E=input('Dose etos')
   A.append(On)
   B.append(E)

c1=0
c2=0
c3=0
for x in range(1800):
   H=2001 - B[x]
   if  H>=0 and H<=25:
      c1+=1
   if  H>=26 and H<=50:
      c2+=1
   if  H>=51:
      c3+=1
   H1.append(H)

print c1 , "NEOI"
print c2 , "MESILIKES"
print c3 , "GERODES"

N=1800
for i in range(1,N,1):
   for j in range (N-1, i-1,-1):
      if H1[j]>H1[j-1]:
         H1[j],H1[j-1]=H1[j-1],H1[j]
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]

for x in range(4):
   print A[x], H1[x]
```

ΑΣΚΗΣΗ 2

```
A=[]
B=[]

for x in range(100):
   On=str(raw_input('Dose Onoma'))
   Xr=input('Dose Xreos')
   A.append(On)
   B.append(Xr)

N=100
for i in range(1,N,1):
   for j in range (N-1, i-1,-1):
      if B[j]>B[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]

for x in range(1,100,1):
   if B[x]==B[0]:
      print A[x]

F=False
P=raw_input('Dose onoma promithefti')
for x in range(100):
   if A[x]== P :
      print B[x]
      F=True
if F==False:
   print "Den parxei o promitheftis"
```

ΑΣΚΗΣΗ 3

```
K=[ ]
ON=[ ]
P=[ ]

for x in range(1000):
   Ko=str(raw_input('Dose Kodiko'))
   On=str(raw_input('Dose Onoma'))
   Pr=input('Dose promithefti')
   K.append(Ko)
   ON.append(On)
   P.append(Pr)

F=False
onoma_P=raw_input('Dose onoma promithefti')
for x in range(1000):
   if P[x]== onoma_P :
      print K[x],ON[x]
      F=True
if F==False:
   print "Den parxei o promitheftis"
```

ΑΣΚΗΣΗ 4

```
N=100
for i in range(1,N,1):
   for j in range (N-1, i-1,-1):
      if A[j]<A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]

for x in range(10):
   print A[x]
```

ΑΣΚΗΣΗ 5

```
A=[]
B=[]
for x in range(100):
   On=str(raw_input('Dose onoma'))
   V=input('Dose vathmo')
   A.append(On)
   B.append(V)

A1=[]
B1=[]
for x in range(100):
   A1.append('-')
   B1.append(-1)

c=0
for x in range(100):
   if B[x] >17.5 :
      B1[c]=B[x]
      A1[c]=A[x]
      c=c+1

N=len(A1)
   for i in range(1,N,1):
      for j in range(N-1, i-1,-1):
         if B1[j] > B1[j-1]:
            B1[j],B1[j-1]= B1[j-1], B1[j]
            A1[j],A1[j-1]= A1[j-1], A1[j]

for x in range(len(A1)):
   print A1[x], B1[x]
```

ΑΣΚΗΣΗ 6

```
def booblesort(A,B,C):
   N=len(A)
   for i in range(1,N,1):
      for j in range (N-1, i-1,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
            C[j],C[j-1]=C[j-1],C[j]
         if A[j]==A[j-1] and B[j] < B[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
            C[j],C[j-1]=C[j-1],C[j]
ON=[]
T=[]
E=[]
N=input('Dose arithmo proiodon')

for x in range(N):
   onoma=str(raw_input('Dose onoma'))
   timi=input('Dose timi')
   eteria=str(raw_input('Dose eteria'))
   ON.append(onoma)
   T.append(timi)
   E.append(eteria)

booblesort(T,ON,E)

F=False
E1=raw_input('Dose onoma eterias')
for x in range(1000):
   if E[x]== E1 :
      print T[x],ON[x]
      F=True
if F==False:
   print "Den parxei h eteria"
```

ΑΣΚΗΣΗ 7

```
def booblesort(A):
   N=len(A)
   for i in range(1,N,1):
      for j in range (N-1, i-1,-1):
         if A[j] > A[j-1]:
               A[j],A[j-1]=A[j-1],A[j]
B=[]
N=[]
A=[]

B1=input('Dose arithmo katastimaton Borias
Elladas')

for x in B1:
   onoma=str(raw_input('Dose onoma'))
   B.append(onoma)

N1=input('Dose arithmo katastimaton Notias
Elladas')

for x in N1:
   onoma=str(raw_input('Dose onoma'))
   N.append(onoma)

booblesort(B)
booblesort(N)

while   B != [ ] and   N != [ ] :
   if   B[0] < N[0] :
      A.append( B.pop(0) )
   else:
      A.append( N.pop(0) )
A=A + B + N
```

ΑΣΚΗΣΗ 8

```
S1=0
for x in P_2019:
   S1=S1 + x

S2=0
for x in P_2020:
   S2=S2 + x
#a
Pososto_Afksisis=100*(S2-S1)/S1

#b

for x in range(27):
   print  A[x], 100*(P_2020[x]-
P_2019[x])/P_2019[x]

#c

max1=0
y1=0

for x in range(27):
   if P_2019[x] > max1:
      max1= P_2019[x]
      y1=x

max2=0
y2=0

for x in range(27):
   if P_2020[x] > max2:
      max2= P_2020[x]
      y2=x

if y1==y2:
   print "kratos me ton max plithismo = ", A[y1]
```

ΑΣΚΗΣΗ 9

```
def booblesort(A, B, C):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1, i-1,-1):
         if C[j] < C[j-1]:
            C[j],C[j-1]= C[j-1], C[j]
            A[j],A[j-1]= A[j-1], A[j]
            B[j],B[j-1]= B[j-1], B[j]
N=input('Dose aritmo pelaton')
A=[]
B=[]
C=[]
for x in range(N):
   On=str(raw_input('Dose onoma'))
   X=input('Dose xrono')
   B.append(On)
   A.append(X)
   if X<=15 and X>=0:
      Xr= X*0.45
   if X >15:
      Xr= 15* 0.45 +(X-15)*0.30
   C.append(Xr)
#a)
for x in range(N):
   print B[x], C[x]
#b)

S=0
for x in C:
   S+=x

K=S* 15/100

#γ
booblesort(A, B, C)

for x in range(10):
   print B[x]
```

ΑΣΚΗΣΗ 10

```
size = int( raw_input(" Μέγεθος αρχείου σε MB = ") )
files = 0
capacity = 500
while  size <= capacity :
     capacity = capacity – size
     files = files + 1
     size = int( raw_input(" Μέγεθος αρχείου σε MB = ") )
print " Αρχεία στο δίσκο = " , files
```

1) Ένα πολυκατάστημα έχει 5 ορόφους. Το ασανσέρ του έχει μέγιστο όριο ασφάλειας βάρους τα 500 κιλά και σύνολο ατόμων 7.

Να γράψετε πρόγραμμα όπου θα δίνετε από το πληκτρολόγιο για κάθε πελάτη που επισκέπτεται το πολυκατάστημα το όνομα του, το βάρος του και τον όροφο που θα επισκεφτεί (1, 2, 3, 4 ή 5). Η εισαγωγή των δεδομένων θα σταματάει όταν δώσετε για όροφο τον αριθμό μηδέν (0). Τα στοιχεία θα καταχωρούνται στις λίστες ON, BA και FL αντίστοιχα.

Θα πρέπει να επισκεφτούν το πολυκατάστημα τουλάχιστον 60 πελάτες.

Α) Πόσες φορές θα ανέβει το ασανσέρ (διαβάζοντας τις λίστες) ώστε να επισκεφτούν όλοι οι πελάτες το πολυκατάστημα (Να μετρηθεί μόνο το ανέβασμα στους ορόφους των πελατών).

Β) Να τυπωθεί το ποσοστό επισκεψιμότητας του κάθε ορόφου.

Γ) Να τυπωθεί το όνομα του πελάτη, με το μικρότερο βάρος, που θα επισκεφτεί τον τελευταίο όροφο ($5^{ος}$)

Σημείωση: Να γίνεται έλεγχος ορθότητας τιμών ώστε το βάρος των πελατών να είναι μεγαλύτερο από το μηδέν

2) Σ’ ένα σχολείο αποφασίστηκε να γίνει συγχώνευση κάποιων τμημάτων για τη διδασκαλία ενός συγκεκριμένου μαθήματος επιλογής. Στο πρώτο τμήμα **Α1** υπάρχουν 12 μαθητές ενώ στο δεύτερο τμήμα **Α2** υπάρχουν 14 μαθητές.

Τα ονόματα των μαθητών είναι καταχωρημένα στις λίστες **Α1** και **Α2** αντίστοιχα.

Επίσης στις λίστες **Β1** και **Β2** αντίστοιχα είναι καταχωρημένοι οι μέσοι όροι των βαθμών του κάθε μαθητή.

Να δημιουργήσετε την συνάρτηση ευθείας ανταλλαγής bubblesort αύξουσα και να την καλέσετε ώστε να ταξινομήσετε ως προς τις βαθμολογίες τις λίστες.

Να δημιουργήσετε κατάλληλη συνάρτηση **Merge1** η οποία θα συγχωνεύσει τα δύο τμήματα σε ένα νέο τμήμα με όνομα Α

Στην συνέχεια να τυπωθούν τα ονόματα των μαθητών με την υψηλότερη βαθμολογία (Δεν είναι μόνο ένα μαθητής με την υψηλότερη βαθμολογία)

3) Ένας σκληρός δίσκος έχει συνολική χωρητικότητα 500 GB. Ο ιδιοκτήτης του ξεκινάει να αποθηκεύει αρχεία στον σκληρό δίσκο, ο οποίος είναι άδειος αρχικά.

Η αποθήκευση θα σταματήσει όταν η χωρητικότητα του δίσκου δεν θα επαρκεί για την αποθήκευση του αρχείου που διαβάζεται.

Να γράψετε πρόγραμμα σε Python το οποίο:

Γ1. Θα διαβάζει επαναληπτικά το όνομα και το μέγεθος (σε GB) του αρχείου που θέλει να αποθηκεύσει ο ιδιοκτήτης, και θα ελέγχει αν μπορεί να αποθηκευτεί στον σκληρό δίσκο.

Γ2. Αν η αποθήκευση μπορεί να πραγματοποιηθεί, θα εμφανίζει το μήνυμα 'Επιτυχής αποθήκευση'. Αν η αποθήκευση δε μπορεί να πραγματοποιηθεί, θα εμφανίζει 'Δεν υπάρχει αρκετός χώρος γι' αυτό το αρχείο.'

Γ3. Στο τέλος θα εμφανίζει:

α. το πλήθος των αρχείων που αποθηκεύτηκαν,

β. το όνομα και το μέγεθος του μεγαλύτερου αρχείου που αποθηκεύτηκε,

γ. το ποσοστό των αρχείων που είχε μέγεθος το πολύ 1 GB,

δ. το μέσο όρο των μεγεθών όλων των αρχείων που αποθηκεύτηκαν στον δίσκο,

ε. τον ελεύθερο χώρο που έμεινε στον σκληρό δίσκο. Αν δεν έχει μείνει χώρος στον σκληρό δίσκο, να εμφανίζει το μήνυμα 'Ο δίσκος έχει γεμίσει.'

Παρατηρήσεις: Θεωρήστε δεδομένο ότι όλα τα αρχεία έχουν διαφορετικό μέγεθος μεταξύ τους και ότι θα επιτευχθεί η αποθήκευση τουλάχιστον ενός αρχείου στο δίσκο.

ΑΣΚΗΣΗ 1

```
ON=[]
BA=[]
FL=[]
fl=input('Dose orofo')
p=1
while fl!=0:
   on=str(raw_input('Dose onoma'))
   ba=input('Dose varos')
   while ba<=0 :
      ba=input('Dose SOSTO varos')
   ON.append(on)
   BA.append(ba)
   FL.append(fl)
   fl=input('Dose orofo')
   while p<60 and fl==0:
      fl=input('Dose orofo diaforetiko apo 0')
   p=p+1

BA1=BA[ : ]

Z=0
while BA1!=[]:
   S=0
   k=0
   M=BA1[0]
   while BA1!=[] and S+M<=500 and k<=7:
      S=S+M
      BA1.pop(0)
      k=k+1
      if BA1!=[]:
         M=BA1[0]
   Z=Z+1
print "Synolika to ansanser anevike ", Z, " fores"

c1=0
c2=0
c3=0
c4=0
c5=0
for x  in FL:
   if x==1:
      c1+=1
   if x==2:
      c2+=1
   if x==3:
      c3+=1
   if x==4:
      c4+=1
```

```
   if x==5:
      c5+=1

print " O 1 orofoe exei pososto episkpsimotitas : ", 100.0*c1/len(FL)
print " O 2 orofoe exei pososto episkpsimotitas : ", 100.0*c2/len(FL)
print " O 3 orofoe exei pososto episkpsimotitas : ", 100.0*c3/len(FL)
print " O 4 orofoe exei pososto episkpsimotitas : ", 100.0*c4/len(FL)
print " O 5 orofoe exei pososto episkpsimotitas : ", 100.0*c5/len(FL)

Vmin=500
for  x  in range(len(ON)):
   if BA[x]<Vmin and FL[x]==5:
      Vmin=BA[x]
      N=ON[x]

print N
```

ΑΣΚΗΣΗ 2 - ΣΥΓΧΩΝΕΥΣΗ

```
def  bubbleSort(A,B):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]

def  Merge1(A1, B1, A2, B2):
   B = []
   A=[]
   while   B1 != [ ] and   B2 != [ ] :
      if   B1[0] < B2[0] :
         B.append( B1.pop(0) )
         A.append( A1.pop(0) )
      else:
         B.append( B2.pop(0) )
         A.append( A2.pop(0) )
   B=B+B1+B2
   A=A+A1+A2
   return A, B

bubbleSort(A1,B1)
bubbleSort(A2,B2)

A, B = Merge1(A1, B1, A2, B2)

for x in range(len(B)):
   if B[x]==B[-1]:
      print A[x]
```

ΑΣΚΗΣΗ 3

```
F=[]
ON=[]
C=500
z=0
f=input('Dose megethos arxeiou')

while f<=C :
   on=str(raw_input('Dose onoma'))
   F.append(f)
   ON.append(on)
   z=z+1
   C=C-f
   print "Epitixis apothikefsi"
   f=input('Dose megethos arxeiou')

print "Den yparxei arketos xoros"


print "plithos arxeion =", z

max1=0
for x  in range(len(F)):
   if  F[x]>max1:
      max1=F[x]
      onoma=ON[x]
print "to arxeio ", onoma, " exei to megalitero megethos =", max1

v=0.0
S=0.0
for  x in F:
   if  x <=1000:
      v=v+1
   S=S+1

print "pososto = ", 100*v/len(F)

print "Mesos oros =", S/len(GF)

if C==0:
   print "den yparxei eleftheros xoros"
else:
   print C
```

1) Στους προκριματικούς αγώνες ιππικού τριάθλου συμμετέχουν 16 αθλητές. Τα αγωνίσματα είναι: «ιππική δεξιοτεχνία», «υπερπήδηση εμποδίων» και «ελεύθερη ιππασία». Ο κάθε αθλητής βαθμολογείται ξεχωριστά σε κάθε ένα από τα τρία αγωνίσματα.

Να γραφεί πρόγραμμα Python το οποίο:

Α. Να τοποθετεί στη λίστα events τις ονομασίες των τριών αγωνισμάτων, όπως αυτές δόθηκαν παραπάνω.

Β. Να διαβάζει και τοποθετεί στις λίστες name, h_name το όνομα αθλητή και το όνομα του αλόγου κάθε αθλητή.

Γ. Να διαβάζει και τοποθετεί στις λίστες v1, v2 και v3 αντίστοιχα τη βαθμολογία κάθε αθλητή σε κάθε αγώνισμα.

Δ. Να διαβάζει το όνομα ενός αθλητή και να εμφανίζει το όνομα του αλόγου με το οποίο αγωνίστηκε και τη συνολική του βαθμολογία στα τρία αγωνίσματα. Αν δεν υπάρχει ο αθλητής, να εμφανίζει κατάλληλα διαμορφωμένο μήνυμα.

Ε. Να εμφανίζει το όνομα (ή τα ονόματα) του αγωνίσματος (ή των αγωνισμάτων) με το μεγαλύτερο «άνοιγμα βαθμολογίας». Ως «άνοιγμα βαθμολογίας» να θεωρήσετε τη διαφορά ανάμεσα στην καλύτερη και στη χειρότερη βαθμολογία του αγωνίσματος.

2) Σε ένα σχολείο φοιτούν 120 μαθητές. Να γραφεί πρόγραμμα Python το οποίο:

Α. Να διαβάζει για κάθε μαθητή το ονοματεπώνυμο του, την τάξη του και τον τελικό βαθμό του και τα καταχωρεί στις αντίστοιχες λίστες NAME, TAKSI, και VATH ελέγχοντας την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:

- Οι τάξεις είναι Α ή Β ή Γ.
- Ο τελικός βαθμός είναι από 1 μέχρι και 20.

Β. Να εμφανίζει τα ονόματα των μαθητών της Β τάξης που έχουν τελικό βαθμό μεγαλύτερο του 18

Γ. Να υπολογίζει και να εμφανίζει το πλήθος των μαθητών κάθε τάξης,

Δ. Να υπολογίζει και εμφανίζει το μέσο όρο των τελικών βαθμών των μαθητών της Γ τάξης και

Ε. Να εμφανίζει ταξινομημένα κατά αλφαβητική σειρά τα ονοματεπώνυμα και τους αντίστοιχους τελικούς βαθμούς των μαθητών της Α τάξης.

ΑΣΚΗΣΗ 1

```
def  MAX_MIN(A):
   max1=A[0]
   min1=A[0]
   for  x  in range(1, len(A)):
      if A[x]> max1:
         max1=A[x]
      if A[x]< min1:
         min1=A[x]
   return max1 - min1

events=[] ; name=[] ; h_name=[] ; V1=[] ; V2=[] ; V3=[] ; D=[]

for  x  in range (3):
   Ag=str(raw_input('Dose onoma agonismatos'))
   events.append(Ag)

for  x  in range(16):
   on=str(raw_input('Dose onoma athliti'))
   on_h==str(raw_input('Dose onoma alogou'))
   name.append(on)
   h_name.append(on_h)
   v1=input('Dose 1h vathmologia')
   v2=input('Dose 1h vathmologia')
   v3=input('Dose 1h vathmologia')
   V1.append(v1)
   V2.append(v2)
   V3.append(v3)

athlitia=raw_input('Dose onoma athliti')
F=False
for  x  in range(16):
   if name[x]== athlitis:
      print name[x], V1[x]+V2[x]+V3[x]
      F=True
if F==False:
   print "Den yparxei o athlitis"

D.append(MAX_MIN(V1))  ;  D.append(MAX_MIN(V2)) ; D.append(MAX_MIN(V3))

max1=D[0]
if D[1]>max1:
   max1=D[1]
if D[2]>max1:
   max1=D[2]

for x  in range(3):
   if D[x]==max1:
      print events[x], D[x]
```

ΑΣΚΗΣΗ 2

```
NAME=[ ]  ; TAKSI=[ ]  ; VATH=[ ]

for  x  in range(120):
   on=str(raw_input('Dose onoma'))
   T=str(raw_input('Dose taksi'))
   while T not in ["A","B","C"]:
      T=str(raw_input('Dose SOSTH taksi'))
   V=input('Dose vathmo')
   while V <0 or V > 20:
      V=input('Dose SOSTO vathmo')

   NAME.append(on)
   TAKSI.append(T)
   VATH.append(V)

for x  in range(120):
   if VATH[x]>18 and TAKSI[x]=="B":
      print NAME[x]

c1=0 ; c2=0;  c3=0
for x  in range(120):
   if TAKSI[x]=="A":
      c1=c1+1
   if TAKSI[x]=="B":
      c2=c2+1
   if TAKSI[x]=="C":
      c3=c3+1
print c1, c2, c3

S=0.0
for x  in range(120):
   if TAKSI[x]=="C":
      S=S+VATH[x]
print S/ c3

AT=[]
VA=[]
for x  in range(120):
   if TAKSI[x]=="A":
      AT.append(NAME[x])
      VA.append(VATH[x])

N=len(AT)
for  i in range(1, N,1):
   for j in range(N-1,i-1,-1):
      if AT[j]<AT[j-1]:
         AT[j],AT[j-1]=AT[j-1],AT[j]
         VA[j],VA[j-1]=VA[j-1],VA[j]

for x  in range(len(AT)):
   print AT[x], VA[x]
```

1) Να δίνετε από το πληκτρολόγιο τον αριθμό των μαθητών του τμήματος **Β1**.

Στην συνέχεια να δίνετε από το πληκτρολόγιο για κάθε μαθητή του Β1 :

Α) το όνομα του

Β) τον βαθμό απολυτηρίου

Τα στοιχεία να τοποθετούνται στις αντίστοιχες λίστες Οn1 και Α1

Στην συνέχεια να δίνετε από το πληκτρολόγιο τον αριθμό των μαθητών του τμήματος **Β2** και για κάθε μαθητή του Β2 να δίνετε από το πληκτρολόγιο:

Α) το όνομα του

Β) τον βαθμό απολυτηρίου

Τα στοιχεία να τοποθετούνται στις αντίστοιχες λίστες Οn2 και Α2

Με τη βοήθεια της συνάρτησης **bubbleSort** να ταξινομήσετε **αύξουσα** ως προς τον βαθμό απολυτηρίου τις λίστες των δύο τμημάτων

Χρησιμοποιώντας τον Αλγόριθμο της συγχώνευσης των στοιχείων δυο ταξινομημένων λιστών σε μία νέα, επίσης ταξινομημένη, λίστα να συνενώσετε τις βαθμολογίες των δύο τμημάτων σε μια νέα ταξινομημένη λίστα.

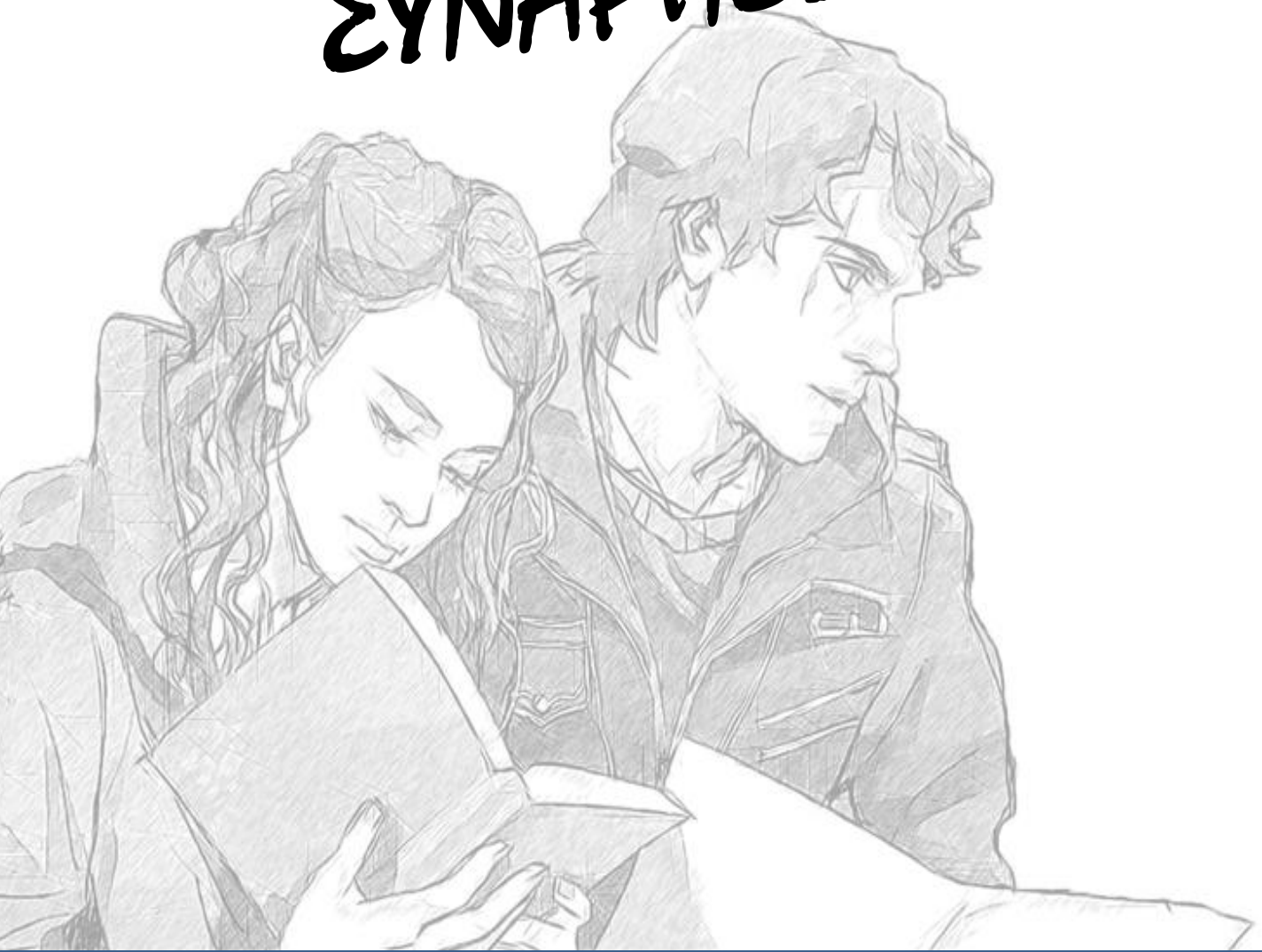
ΑΣΚΗΣΗ 1

```
def  bubbleSort(A,B):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1, i-1,-1):
         if B[j]<B[j-1]:
            B[j], B[j-1]=B[j-1],B[j]
            A[j], A[j-1]=A[j-1],A[j]
SOn=[]
SA=[]

On1=[]
A1=[]
On2=[]
A2=[]
N1=input('Dose # mathiton B1')
N2=input('Dose # mathiton B1')
for  x  in range (N1):
   onoma=str(raw_input('Dose onoma'))
   V=input('Dose vathmo')
   On1.append(onoma)
   A1.append(V)

for  x  in range (N2):
   onoma=str(raw_input('Dose onoma'))
   V=input('Dose vathmo')
   On2.append(onoma)
   A2.append(V)

bubbleSort(On1,A1)
bubbleSort(On2,A2)

while   A1 != [ ] and   A2 != [ ] :
   if   A1[0] < A2[0] :
      SA.append( A1.pop(0) )
      SOn.append(On1.pop(0))
   else:
      SA.append( A2.pop(0) )
      SOn.append(On2.pop(0))

SA=SA + A1 + A2
SOn=SOn= On1 + On2
```

ΚΕΦΑΛΑΙΟ 7

ΠΡΟΗΓΜΕΝΑ ΣΤΟΙΧΕΙΑ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

ΣΥΝΑΡΤΗΣΕΙΣ



ΑΣΚΗΣΕΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

ΠΑΡΑΔΕΙΓΜΑ

```
def  A1():
   return 10 , 20, 30
x1, x2, x3 =A1()
print x1                # θα τυπώσει  10
print x2                # θα τυπώσει  20
print x3                # θα τυπώσει  30
```

1) Η βιβλιοθήκη του δήμου, θέλει να οργανώσει τα διαθέσιμα βιβλία της ηλεκτρονικά.

Διαθέτει 3000 βιβλία τα οποία θα αποθηκεύσει στην λίστα Β.

Επιπρόσθετα, στην λίστα S θα αποθηκεύσει το όνομα του Συγγραφέα.

Η βιβλιοθήκη καταχωρεί σε μια λίστα ΟΝ το ονοματεπώνυμο των δανειστών της οι οποίοι είναι 1000. Ο δανεισμός των βιβλίων είναι μηνιαίος.

Τέλος σε μια λίστα D καταχωρείται πόσες φορές έχει γίνει δανεισμός του κάθε βιβλίου.

Να γίνουν:

Α. υποπρόγραμμα το οποίο θα διαβάζει τις προαναφερθέντες λίστες από το πληκτρολόγιο του χρήστη και θα τις επιστρέφει στο κύριο πρόγραμμα.

Β. υποπρόγραμμα το οποίο θα βρίσκει, και θα εμφανίζει για πιο βιβλίο (όνομα και συγγραφέας) έχουν γίνει οι περισσότεροι δανεισμοί.

Γ. υποπρόγραμμα το οποίο θα διαβάζει το όνομα ενός συνδρομητή και να εμφανίζει αν υπάρχει στην λίστα.

ΑΣΚΗΣΗ 1

```
B=[] ; S=[] ; D=[] ; ON=[]
def   Eisagogi():
        B=[] ; S=[] ; D=[] ; ON=[]
        for    x  in range (3000):
                Book=str(raw_input('Dose Onoma Vivliou)
                Onoma=str(raw_input('Dose Onoma Sygrafea')
                N=input('Dose arithmo danismou')
                B.append(Book)
                S.append(Onoma)
                D.append(N)
        for    y  in range (1000):
                danismos=str(raw_input('Dose Onoma daniston)
                ON.append(danismos)
        return  B, S, D, ON
def   spy(D, B, S):
        max1=0
        y=0
        for  x   in range(len(D)):
                if  D[x] > max1:
                        max1=D[x]
                        y=x
        print B[y], S[y], D[y]
def  FindOn(ON, Name):
        if  Name  in  ON:
                print 'Yparxei'
        else :
                print 'Den Yparxei'
B, S,D, ON =Eisagogi()
spy(D, B, S)
Name= raw_input('Dose Onoma')
FindOn(ON, Name)
```

1) Να γραφεί πρόγραμμα που για 50 οικογένειες θα διαβάζει σε λίστα τον αριθμό παιδιών της, δεχόμενοι ότι τα παιδιά μπορεί να είναι από 1 μέχρι και 10 ανά οικογένεια.

Να γραφεί

α) υποπρόγραμμα το οποίο θα διαβάζει τις προαναφερθέντες λίστες από το πληκτρολόγιο του χρήστη και θα τις επιστρέφει στο κύριο πρόγραμμα. Για κάθε οικογένεια θα διαβάζεται το επίθετό της και τον αριθμό των παιδιών της.

β) Να εμφανιστούν στο πρόγραμμα τα επίθετα των οικογενειών και ο αριθμός παιδιών σε φθίνουσα σειρά ταξινόμησης με βάση τον αριθμό παιδιών. Σε περίπτωση ισότητας στον αριθμό των παιδιών μεταξύ διαδοχικών οικογενειών, να εμφανίζονται τα ονόματα των οικογενειών αλφαβητικά.

ΑΣΚΗΣΗ 1

```
A=[]
B=[]


def  Eisagogi():
   A1=[]
   B1=[]
   for  x  in range(50):
      E=str(raw_input('Dose Onoma'))
      P=input('Dose arithmo paidion')
      while  P<1  or  P>10:
         P=input('Dose arithmo paidion')
      A1.append(E)
      B1.append(P)
   return  A1, B1

A, B  = Eisagogi()

N=len(B)
for  i  in range (1, N, 1):
   for   j in range (N-1, i-1, -1):
      if  B[j] > B[j-1]:
         B[j], B[j-1]= B[j-1], B[j]
         A[j], A[j-1]= A[j-1], A[j]



for  i  in range (1, N, 1):
   for   j in range (N-1, i-1, -1):
      if  B[j] == B[j-1]  and  A[j]<A[j-1]:
         B[j], B[j-1]= B[j-1], B[j]
         A[j], A[j-1]= A[j-1], A[j]

for  x  in range (50):
   print  A[x], B[x]
```

1) Να γράψετε συναρτήσεις οι οποίες θα εκτελούν τις ίδιες λειτουργίες με τις παρακάτω γνωστές συναρτήσεις:

A) len(), B) max() , C) min(), D) abs(), E) pow()

2) Να γράψετε συνάρτηση η οποία να δέχεται για είσοδο μια συμβολοσειρά και ένα χαρακτήρα και να μας επιστρέφει πόσες φορές εμφανίζεται ο χαρακτήρας μέσα στην συμβολοσειρά

3) Να γράψετε συνάρτηση η οποία να δέχεται για είσοδο μια συμβολοσειρά και ένα χαρακτήρα και να μας επιστρέφει την θέση του χαρακτήρα μέσα στην συμβολοσειρά

4) Να γράψετε συνάρτηση η οποία να δέχεται για είσοδο μια συμβολοσειρά και να μας επιστρέφει τον αριθμό των φωνηέντων της συμβολοσειράς.

5) Να σχηματίσετε πίνακα τιμών του παρακάτω προγράμματος

```
def A1(C):
    global  A
    global  B
    A=A-2
    B=B+5
    C=C+A

A=3
B=13
C=2
print A,B,C
A1(C)
print A,B,C
A1(C)
print A,B,C
```

A	**B**	**C**	**print A,B,C**

6) Να γράψετε συνάρτηση με όνομα compare η οποία να δέχεται για είσοδο δύο αριθμούς και να τους συγκρίνει και να τυπώνει ποιος είναι μεγαλύτερος ή αν είναι ίσοι μεταξύ τους.

7) Σε ένα αρχείο με όνομα A.txt έχουν αποθηκευτή τα ονόματα και ο βαθμός 1^{ου} τετράμηνου 20 μαθητών της Α τάξης. Κάθε εγγραφή είναι σε ξεχωριστή γραμμή όπως το παράδειγμα:

Maria
18
Nikos
16
Petros
20
Να γράψετε συνάρτηση η οποία να δέχεται για είσοδο το όνομα του αρχείου και να επιστρέφει το μέσο όρο της βαθμολογίας των μαθητών

1A	**1B**	**1C**

def len1(A):
  c=0
  for x in A:
  c=c+1
  return c
	def MAX1(A):
	  max1=A[0]
	  for x in range (1, len(A), 1):
	  if A[x]> max1:
	  max1=A[x]
	  return max1
		def MIN1(A):
		  min1=A[0]
		  for x in range (1, len(A), 1):
		  if A[x]< min1:
		  min1=A[x]
		  return min1

1D	**1E**	**2**

def ABS1(x):
 	
  if x <0:
  x=-x
 	
  return x
	def POW1(x,y):
	
	  return x**y
		def A1(A,y):
		  c=0
		  for x in A:
		  if x==y:
		  c=c+1
		
		  return c

3	**4**	**6**

def A1(A,y):
 	
  for x in range(len(A)):
  if A[x]==y:
  z=x
 	
  return z
	def A1(A):
	  c=0
	  F="AEOIHYaeoihy"
	  for x in A:
	  if x in F:
	  c=c+1
	
	  return c
		def compare (x,y):
		  if x>y:
		  return x
		  if y>x:
		  return y
		  if x==y:
		  return "isoi"

7		

def A1 (Onoma):
  f1=open(Onoma,"r")
  c=1
  S=0
  c1=0.0
  for x in f1:
  if c%2==0:
  S=S+int(x)
  c1=c1+1
  c=c+1
  return S/c1		

5

A	**B**	**C**	**print A,B,C**
3	13	2	3 13 2
1	18	3	1 18 2
-1	23	1	-1 23 2

1) Σε μια εταιρεία υπάρχουν 3 κατηγορίες υπαλλήλων με κωδικούς Κ1, Κ2 και Κ3 αντίστοιχα. Να γραφτεί πρόγραμμα που θα διαβάζει για κάθε υπάλληλο τον κωδικό του και το μηνιαίο μισθό του σε Ευρώ

Α) να γραφτεί συνάρτηση η οποία να υπολογίζει και να επιστρέφει πόσοι υπάλληλοι υπάρχουν σε κάθε κατηγορία

Β) Οι υπάλληλοι θα πάρουν δώρο με βάση τον παρακάτω πίνακα:

ΚΑΤΗΓΟΡΙΑ	**ΔΩΡΟ**
Κ1	1/10 του μισθού
Κ2	1/15 του μισθού
Κ3	30€

Να γραφτεί κώδικας που θα υπολογίζει και θα τυπώνει τον μισθό κάθε υπαλλήλου

Γ) Να γραφτεί συνάρτηση που θα υπολογίζει και θα τυπώνει τον μέσο όρο των μισθών όλων των υπαλλήλων της Κ3 κατηγορίας

Σημείωση:

Η εισαγωγή δεδομένων θα σταματήσει όταν δοθεί ως είσοδος μη αποδεκτός κωδικός

Οι μισθοί πρέπει να είναι μεταξύ 800€ με 1500€

2) Να γραφεί πρόγραμμα όπου θα δίνουμε από το πληκτρολόγιο:

Τον τύπο του αυτοκινήτου που περάνει από μια διασταύρωση (Ι για ΙΧ, Μ για μοτοσυκλέτα και F για φορτηγό)

Το χρώμα του αυτοκινήτου (αποδεκτές τιμές Άσπρο, Μαύρο, Κόκκινο και Μπλε)

Να γραφτεί συνάρτηση η οποία θα αποθηκεύει:

- στο αρχείο Α1.txt τον αριθμό των Άσπρων, Μαύρων, Κόκκινων και Μπλε ιδιωτικής χρήσης οχημάτων
- στο αρχείο Α2.txt τον αριθμό των Μαύρων και Κόκκινων μοτοσυκλετών
- στο αρχείο Α3.txt τον αριθμό των Μπλε φορτηγών

Σημείωση:

Η εισαγωγή δεδομένων θα σταματήσει όταν δοθεί ως τύπος αυτοκινήτου η λέξη «TELOS»

ΑΣΚΗΣΗ 1

```
def Synolo(A):
   c1=0
   c2=0
   c3=0
   for  x  in A:
      if x=="K1":
         c1=c1+1
      if x=="K2":
         c2=c2+1
      if x=="K3":
         c3=c3+1
   return c1, c2, c3


def  Mesos_Oros(A,B):
   v=0.0
   S=0
   for x  in range(len(A)):
      if A[x]=="K3":
         S=S+B[x]
         v=v+1
   print S/v

A=[]
B=[]

K=str(raw_input('Dose katigoria'))
while  K=="K1" or K=="K2"  or K=="K3":
   M=input('Dose Mistho')
   while M<800 or M>1500:
      M=input('Dose Mistho')
   A.appand(K)
   B.append(M)
   K=str(raw_input('Dose katigoria'))

c1, c2, c3=Synolo(A)
print c1, c2, c3

for  x  in range(len(B)):
   if A[x]=="K1":
      print  B[x]+B[x]/10
   if A[x]=="K2":
      print  B[x]+B[x]/15
   if A[x]=="K3":
      print  B[x]+30

Mesos_Oros(A,B)
```

ΑΣΚΗΣΗ 2

```
def   Save_cars(A,B):

   c1=0
   c2=0
   c3=0

   f1=open("A1.txt","w")
   for x in range(len(A)):
      if A[x]=="I":
         c1=c1+1
   f1.write(str(c1) + "\n")
   f1.close()

   f2=open("A2.txt","w")
   for x in range(len(A)):
      if A[x]=="M":
         if B[x]=="Mavro" or B[x]=="Kokkino":
            c2=c2+1
   f2.write(str(c2) + "\n")
   f2.close()

   f3=open("A3.txt","w")
   for x in range(len(A)):
      if A[x]=="F":
         if B[x]=="Mple" :
            c3=c3+1
   f3.write(str(c3) + "\n")
   f3.close()


A=[]
B=[]
F=["Aspro", "Mavro", "Kokkino", "Mple"]
T=str(raw_input("Dose tipo"))
while  T!='TELOS':
   X=str(raw_input('Dose xroma'))
   while  X not in F:
      X=str(raw_input('Dose SOSTO xroma'))

   A.append(T)
   B.append(X)

   T=str(raw_input("Dose tipo"))

Save_cars(A,B)
```

1)

Να γράψετε μια συνάρτηση η οποία θα δέχεται από το κυρίως πρόγραμμα τις ώρες εργασίας και τον ωριαίο μισθό του κάθε υπαλλήλου. Η διαδικασία να υπολογίζει και να επιστρέφει στο κυρίως πρόγραμμα τα εξής:
ι) τον ακάθαρτο μισθό του υπαλλήλου (πολλαπλασιάζονται οι ώρες εργασίας με την ωριαία αμοιβή)
ιι) το φόρο εισοδήματος που υπολογίζεται 20% πάνω στον ακάθαρτο μισθό
ιιι) τις κοινωνικές ασφαλίσεις που υπολογίζονται 6% πάνω στον ακάθαρτο μισθό
Να γράψετε ένα πρόγραμμα το οποίο θα δέχεται το όνομα, τις ώρες εργασίας και την ωριαία αμοιβή 20 υπαλλήλων. Στη συνέχεια κάνοντας χρήση της πιο πάνω συνάρτησης να υπολογίζει τον καθαρό μισθό του κάθε υπαλλήλου. Το πρόγραμμα να παρουσιάζει το όνομα του κάθε υπαλλήλου, τον ακάθαρτο του μισθό, το φόρο εισοδήματος, τις κοινωνικές ασφαλίσεις καθώς και τον καθαρό του μισθό.

2)

Μια εταιρεία έχει 48 υπαλλήλους και τους πληρώνει κάθε εβδομάδα σύμφωνα με τον παρακάτω πίνακα κλιμακωτά.

ΩΡΕΣ ΕΡΓΑΣΙΑΣ	**€/ΩΡΑ**
από 0 έως και 40	5
από 41 έως και 50	7
πάνω από 50	9

Για παράδειγμα, ένας υπάλληλος που δούλεψε 45 ώρες σε μία εβδομάδα θα πληρωθεί 235 €.
Να αναπτύξετε πρόγραμμα στη γλώσσα προγραμματισμού Python το οποίο:
α). Να διαβάζει το όνομα του κάθε υπαλλήλου.
β). Να διαβάζει τον αριθμό των ωρών εργασίας του κάθε υπαλλήλου σε μία εβδομάδα, ελέγχοντας ότι είναι αριθμός μεγαλύτερος ή ίσος του μηδενός (έλεγχος αποδεκτών τιμών).
γ). Να υπολογίζει με τη χρήση συνάρτησης την εβδομαδιαία αμοιβή του κάθε υπαλλήλου.
δ). Να εμφανίζει το όνομα και την εβδομαδιαία αμοιβή του κάθε υπαλλήλου.
ε). Να υπολογίζει και να εμφανίζει το συνολικό κόστος για την εταιρεία.
στ). Να υπολογίζει και να εμφανίζει το ποσοστό των εργαζομένων που πήραν μισθό μεγαλύτερο από 700€.
η). Να γράψετε τη συνάρτηση υπολογισμού της εβδομαδιαίας αμοιβής του υπαλλήλου.

3)

Να γράψετε πρόγραμμα σε Python το οποίο μέσω συνάρτησης θα διαβάζει έναν βαθμό ο οποίος θα πρέπει να είναι στο διάστημα από 1 έως 20 σε περίπτωση που δοθεί λάθος βαθμός θα εμφανίζει κατάλληλο μήνυμα. Η συνάρτηση θα πρέπει να επιστρέφει την τιμή στο κυρίως πρόγραμμα.

4)

Να αναπτυχθεί υποπρόγραμμα που δέχεται το βασικό μισθό ενός υπαλλήλου και τον αριθμό των παιδιών του και επιστρέφει το επίδομα που αυτός δικαιούται για τα παιδιά του. Το επίδομα υπολογίζεται ως εξής: Για τα δύο πρώτα παιδιά είναι 3% για το κάθε παιδί επί του μισθού. Για το τρίτο παιδί είναι 5% επί του μισθού. Για κάθε επιπλέον παιδί είναι 8% επί του μισθού. Με ποιο τρόπο θα κληθεί το παραπάνω υποπρόγραμμα μέσα από το κυρίως πρόγραμμα.

ΑΣΚΗΣΗ 1

```
def  misthos(Or, M):
   Ma=Or*M
   Foros=Ma*20/100.0
   Ka=Ma*6/100.0

   return Ma, Foros, Ka


for x in range (20):
   Onoma=raw_input("Dose onoma")
   Or=input("Dose Ores ergasias")
   M=input("Dose orieo mistho")

   Ma, Foros, Ka = misthos(Or, M)

   Kathsros_Misthos = Ma - Foros- Ka

   print Onoma, Ma, Foros, Ka, Kathsros_Misthos
```

ΑΣΚΗΣΗ 2

```
def  Amivi(Or):
   if Or >=0 and Or <= 40:
      Xr= Or*5
   if Or >=41 and Or <= 50:
      Xr=40*5 + (Or - 40)*7
   if Or >50:
      Xr=40*5 + 10*7 + (Or - 50)*9

   return Xr

S=0
c=0
for x in range (48):
   Onoma=raw_input("Dose onoma")

   Or=input("Dose Ores ergasias")
   while Or<0:
      Or=input("Dose Sostes Ores ergasias")

   print "O ", Onoma, " exei evdomadiea amivi = ", Amivi(Or)

   S=S+ Amivi(Or)

   if Amivi(Or) >700 :
      c=c+1


Pososto=100*c/48.0

print "Sinoliko kostos = ", S
print "Pososto pano apo 700 = ",Pososto
```

ΑΣΚΗΣΗ 3

```
def  Elenxos():
   V=input("Dose vathmo")
   while V<0 or V>20:
      print "Dose sosto Vathmo"
   return V

print Elenxos()
```

ΑΣΚΗΣΗ 4

```
def  Epidoma(M, P):
   if P>=1 and P<=2:
      E= P * M*3/100.0
   if P==3:
      E= 2 * M*3/100.0 + (P-2) * M*5/100.0
   if P>3:
      E= 2 * M*3/100.0 +  M*5/100.0 + (P-3) * M*5/100.0

   return E

M=input("Dose Mistho")
P=input("Dose arithmo pedion")

print Epidoma(M, P)
```

1) Να δημιουργήσετε μια συνάρτηση με όνομα “Athrisma” η οποία να δέχεται για είσοδο τρείς αριθμούς x, y, z και θα επιστρέφει το άθροισμα τους (x+y+z).

2) Να δημιουργήσετε μια συνάρτηση με όνομα “Α1” η οποία θα δέχεται για είσοδο μια λίστα Β . Η συνάρτηση θα υπολογίζει και θα τυπώνει πόσοι αριθμοί είναι μεταξύ 10 και 20.

Στο κυρίως πρόγραμμα :

Να γεμίσετε την λίστα Β με αριθμούς μέχρι να πληκτρολογήσετε μηδέ. Στην συνέχεια να καλέσετε την συνάρτηση Α1

3) Να δημιουργήσετε μια συνάρτηση με όνομα “ΔΕΗ” η οποία να δέχεται για είσοδο την κατανάλωση ρεύματος και να υπολογίζει και να επιστρέφει το ποσό που θα πληρωθεί με βάση την κλιμακωτή χρέωση :

Κατανάλωση Kwh	Χρήματα ανά Kwh
1 έως και 20	0,50€
21 εως και 50	0,80€
Πάνω από 50	0,90€

Στο κυρίως πρόγραμμα:

Α) να δίνετε από το πληκτρολόγιο τον αριθμό Ν των πελατών

Β) Για κάθε πελάτη να δίνετε από το πληκτρολόγιο την κατανάλωση ρεύματος

Γ) να καλείτε την συνάρτηση ΔΕΗ

ΑΣΚΗΣΗ 1

```
def  Athrisma (x,y,z):
   return  x+y+z
```

ΑΣΚΗΣΗ 2

```
def  A1(B):
   c=0
   for x in B:
      if x >=10 and x<=20:
         c=c+1
   print c

B=[]
y=input('Dose arithmo')
while y !=0:
   B.append(y)
   y=input('Dose arithmo')
A1(B)
```

ΑΣΚΗΣΗ 3

```
def DEH(K):
   if  K>=1 and K<=20:
      Xr=K*0.50
   if  K>=21 and K<=50:
      Xr=20*0.50 + (K-20)*0.80
   if  K>50:
      Xr=20*0.50 + 30*0.80+ (K-50)*0.90

   return  Xr

N=input('Dose arithmo pelaton')

for x in range(N):
   K=input('Dose katakalosi revmatos')
   print DEH(K)
```

1) Να αναπτυχθεί υποπρόγραμμα που δέχεται το βασικό μισθό ενός υπαλλήλου και τον αριθμό των παιδιών του και **επιστέφει** το επίδομα που αυτός δικαιούται για τα παιδιά του. Το επίδομα υπολογίζεται ως εξής: Για τα δύο πρώτα παιδιά είναι 3% για το κάθε παιδί επί του μισθού. Για το τρίτο παιδί είναι 5% επί του μισθού. Για κάθε επιπλέον παιδί είναι 8% επί του μισθού.

2) Το Υπουργείο Περιβάλλοντος αποφάσισε να παρακολουθεί για τριάντα (30) ημέρες τα επίπεδα ενός ρίπου στην ατμόσφαιρα, πραγματοποιώντας μία μέτρηση την ημέρα. Έχουν καθοριστεί τρία επίπεδα μόλυνσή με βάση την τιμή του ρίπου, όπως φαίνεται στον παρακάτω πίνακα:

Τιμές Ρίπου	**Επίπεδα μόλυνσης**
Έως και 1	Φυσιολογικό
Άνω από 1 έως και 2	Οριακό
Πάνω από 2	Επικίνδυνο

Να γράψετε ένα πρόγραμμα σε Python το οποίο:

α. Για κάθε μία από τις τριάντα (30) ημέρες ενός μήνα να διαβάζει την τιμή του ρίπου με τη χρήση κατάλληλου μηνύματος (δε χρειάζεται να γίνεται έλεγχος ορθότητας τιμών).

β. Να εμφανίζει για κάθε μέρα το επίπεδο μόλυνσή ανάλογα με την τιμή του ρίπου.

γ. Να υπολογίζει και να εμφανίζει το πλήθος των ημερών κατά τη διάρκεια των οποίων η τιμή του ρίπου ξεπέρασε την τιμή 3.

δ. Να υπολογίζει και να εμφανίζει τον μέσο όρο των τιμών του ρίπου για το διάστημα των τριάντα (30) ημερών.

3) Να γράψετε μια συνάρτηση η οποία να δέχεται από το κυρίων πρόγραμμα τις ημερες εργασία και τον ωριαίο μισθό του κάθε υπαλλήλου. Η διαδικασία να υπολογίζει και να επιστρέφει στο κυρίων πρόγραμμα τον **ακαθάριστο μισθό του υπαλλήλου** (πολλαπλασιάζονται οι ημέρες εργασίας με την ωριαία αμοιβή)

Να γράψετε μια συνάρτηση η οποία να δέχεται για είσοδο τον ακαθάριστο μισθό και να επιστρέφει το φόρο εισοδήματος που υπολογίζεται **20%** πάνω στον ακαθάριστο μισθό

Να γράψετε μια συνάρτηση η οποία να δέχεται για είσοδο τον ακαθάριστο μισθό και να επιστρέφει τις κοινωνικές ασφαλίσεις που υπολογίζονται **6%** πάνω στον ακαθάριστο μισθό

Να γράψετε ένα πρόγραμμα στο οποίο να πληκτρολογείτε το όνομα, τις ημέρες εργασίας και την ωριαία αμοιβή 20 υπαλλήλων.

Στη συνέχεια κάνοντας χρήση των πάνω συναρτήσεων να υπολογίζει τον καθαρό μισθό του κάθε υπαλλήλου (Ακαθάριστος μείον τον φόρο εισοδήματος μείον τις κοινωνικές ασφαλίσεις).

Το πρόγραμμα να παρουσιάζει το όνομα του κάθε υπαλλήλου, τον ακαθάριστο του μισθό, το φόρο εισοδήματος, τις κοινωνικές ασφαλίσεις καθώς και τον καθαρό του μισθό.

ΑΣΚΗΣΗ 1

```
def  Epidoma(M,P):
   if P>=1 and P<=2:
      E= P*M*3/100
   if P==3:
      E= 2*M*3/100 + M*5/100
   if P>3:
      E= 2*M*3/100 + M*5/100 + (P-3)*M*8/100
```

ΑΣΚΗΣΗ 2

```
S=0.0
c=0
for x in range(30):
   R=input('Dose timi ripou')
   if R>=0 and R<=1:
      print "Fisiologiko"
   if R>1 and R<=2:
      print "Oriako"
   if R >2:
      print "Epikindino"

   if R>3:
      c=c+1

   S=S+R
print c
MO=S/30
print MO
```

ΑΣΚΗΣΗ 3

```
def  Akatharistos(H, M):
   return H*M

def  Foros(Ak):
   return Ak*20/100

def  KA(Ak):
   return  Ak*6/100


for  x in range (20):
   On=raw_input('Dose onoma')
   H=input('Dose hmeres')
   M=input('Dose Mistho')
   Ak=Akatharistos(H, M)
   Katharos=Ak-Foros(Ak) - KA(Ak)
   print On, Foros(Ak), KA(Ak), Ak, Katharos
```

1. Να δημιουργήσετε μια δική σας συνάρτηση η οποία θα κάνει ότι και η συνάρτηση **len(A)** για μια λίστα Α.

2. Να δημιουργήσετε μια δική σας συνάρτηση η οποία θα κάνει ότι και η συνάρτηση **abs(x).**

3. Να δημιουργήσετε μια συνάρτηση η οποία να δέχεται για είσοδο μια λίστα Α και θα επιστρέφει τον αριθμό των αρνητικών αριθμών της λίστας Α δηλαδή πόσους αρνητικούς αριθμούς περιέχει η λίστα.

4. Να δημιουργήσετε μια συνάρτηση η οποία θα δέχεται για είσοδο μια συμβολοσειρά και κάθε γράμμα της συμβολοσειράς θα το εισάγει σε μια λίστα. Τέλος θα τυπώνει την λίστα.

5. Να δημιουργήσετε μια συνάρτηση η οποία θα δέχεται για είσοδο μια λίστα και θα την ταξινομεί **από τον μεγαλύτερο προς τον μικρότερο αριθμό**. Να επιστρέφει την λίστα.

6. Να δημιουργήσετε μια συνάρτηση η οποία να δέχεται για είσοδο έναν αριθμό και μια λέξη και να τυπώνει την λέξη τόσες φορές όσος είναι ο αριθμός που δώσατε για είσοδο. Δηλαδή αν δώσουμε 3 και «Poli» θα τυπώσει Poli Poli Poli

7. Να δημιουργήσετε μια συνάρτηση η οποία να δέχεται για είσοδο τις δικαιολογημένες και τις αδικαιολόγητες απουσίες ενός μαθητή. Να υπολογίζει και να επιστρέφει το σύνολο των απουσιών.

Στο κυρίως πρόγραμμα να ελέγχει αν ο μαθητής έχει πάνω από 114 απουσίες τότε να τυπώνει το μήνυμα «Πρόβλημα».

8. Να ορίσετε μια συνάρτηση με όνομα M_O η οποία να δέχεται για είσοδο μια λίστα και να υπολογίζει και να επιστρέφει τον μέσο όρο των στοιχείων της λίστας

9. Να ορίσετε μια συνάρτηση με όνομα MAX1 η οποία να δέχεται για είσοδο μια λίστα και να επιστρέφει το μεγαλύτερο στοιχείο της λίστας.

ΑΣΚΗΣΗ 1

```
def  LEN(A):
   c=0
   for x in A:
      c+=1
   return c
```

ΑΣΚΗΣΗ 2

```
def  ABS(n):
   if  n<0:
      return -n
   else:
      return n
```

ΑΣΚΗΣΗ 3

```
def  A1(A):
   c=0
   for x in A:
      if x<0:
         c=c+1
   return  c
```

ΑΣΚΗΣΗ 4

```
def  LIST(A):
   B=[]
   for x in A:
      B.append(x)

   print B
```

ΑΣΚΗΣΗ 5

```
def  Fisalida(A):


   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if A[j]>A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
   return A
```

ΑΣΚΗΣΗ 6

```

def   Leksi(A,n):
   print A*n
```

ΑΣΚΗΣΗ 7

```
def Apousies(D,A):
   S=A+D
   print S
   if S>114:
      print 'Provlima'
```

ΑΣΚΗΣΗ 8

```
def M_O(A):
   S=0.0
   for x in A:
      S=S+x
   return S/len(A)
```

ΑΣΚΗΣΗ 9

```

def MAX1(A):
   max1=A[0]
   for x in range(1, lemn(A),1):
      if A[x]>max1:
         max1=A[x]
   return(max1)
```

1) Να δημιουργήσετε συνάρτηση η οποία θα τυπώνει την φράση «Ελευθερία ή Θάνατος»

2) Να δημιουργήσετε συνάρτηση με όνομα Aferesi η οποία θα δέχεται για είσοδο δύο αριθμούς και θα τυπώνει την διαφορά τους.

Στο κυρίως πρόγραμμα θα δίνετε από το πληκτρολόγιο τους δύο αριθμούς

3) Να δημιουργήσετε συνάρτηση με όνομα Mesos_Oros η οποία θα δέχεται για είσοδο τους βαθμούς στα 4 μαθήματα των πανελλαδικών και θα υπολογίζει και θα τυπώνει τον μέσο όρο τους. Αν ο μέσος όρος είναι μικρότερος από 10 θα τυπώνει «Δεν τα κατάφερε»

Στο κυρίως πρόγραμμα θα δίνετε από το πληκτρολόγιο τους 4 βαθμούς

4) Να δημιουργήσετε συνάρτηση με όνομα Megistos η οποία θα δέχεται για είσοδο 4 αριθμούς και θα υπολογίζει και θα επιστρέφει τον μεγαλύτερο από τους τέσσερεις.

Στο κυρίως πρόγραμμα θα δίνετε από το πληκτρολόγιο τους 4 αριθμούς και θα τυπώνετε στην οθόνη ο μεγαλύτερος αριθμός που θα επιστρέψει η συνάρτηση

5) Στο κυρίως πρόγραμμα θα δίνετε από το πληκτρολόγιο τον αριθμό των μαθητών μιας τάξης Ν.

Να δημιουργήσετε συνάρτηση με όνομα Apousies η οποία θα δέχεται για είσοδο τον αριθμό των μαθητών. Στην συνέχεια για κάθε μαθητή να δίνετε από το πληκτρολόγιο τον αριθμό των δικαιολογημένων και των αδικαιολόγητων απουσιών.

Θα υπολογίζει και θα τυπώνει το σύνολο των απουσιών.

Αν το σύνολο είναι μικρότερο ή ίσο από 114 τότε θα τυπώνετε το μήνυμα «όλα καλά» αλλιώς «πρόβλημα»

ΑΣΚΗΣΗ 1

```
def  print1():
   print "Eleftheria h thanatos"

print1()
```

ΑΣΚΗΣΗ 2

```
def  aferesi(x,y):
   print x-y

x=input('Dose x ')
y=input('Dose y ')

aferesi(x,y)
```

ΑΣΚΗΣΗ 3

```
def  Mesos_Oros(v1, v2, v3, v4):
   MO= (v1+v2+v3+ v4)/4.0
   if MO < 10 :
      print 'den ta katafere'

v1=input('Dose v1 ')
v2=input('Dose v2 ')
v3=input('Dose v3 ')
v4=input('Dose v4 ')
Mesos_Oros(v1, v2, v3, v4)
```

ΑΣΚΗΣΗ 4

```
def  Megistos(v1, v2, v3, v4):
   MAX=v1
   if v2>MAX:
      MAX=v2
   if v3>MAX:
      MAX=v3
   if v4>MAX:
      MAX=v4
   return MAX

v1=input('Dose v1 ')
v2=input('Dose v2 ')
v3=input('Dose v3 ')
v4=input('Dose v4 ')
print Megistos(v1, v2, v3, v4)
```

ΑΣΚΗΣΗ 5

```
def  Apousies(N):

   for x in range (N):

      D=input('Dose dikeologimenes ')
      A=input('Dose adikeologites ')
      S=D+A
      if S<=114 :
         print 'Ola kala'
      else:
         print 'Diskola'

N=input ("Dose arithmo mathiton")
Apousies(N)
```

1 - Δραστηριότητα 1

Να εντοπίσετε στο παρακάτω πρόγραμμα τα τμήματα κώδικα που πρέπει να γίνουν συναρτήσεις και να το ξαναγράψετε, έτσι ώστε να αποφευχθεί η επανάληψή τους και να μειωθεί ο όγκος του.

```
sum1 = 0

for i in range(100):

        sum1 = sum1 + i

print sum1

sum2 = 0

for j in range(100):

        sum2 = sum2 + j

print sum2 + sum1

sum3 = 0

for k in range(sum1):

        sum3 = sum3 + k

print sum3 + sum2 + sum1
```

2 - Δραστηριότητα 2

Να ορίσετε μια συνάρτηση product η οποία δέχεται μια λίστα ακεραίων και επιστρέφει το γινόμενο όλων των στοιχείων της λίστας.

3 - Δραστηριότητα 3

Να γράψετε ένα πρόγραμμα σε Python το οποίο

Θα γεμίζετε μια ουρά με αριθμούς μέχρι να πληκτρολογήσετε 0.

θα υπολογίζει το μέσο όρο των θετικών αριθμών.

Για την εισαγωγή των αριθμών και τον υπολογισμό του μέσου όρου, να υλοποιήσετε κατάλληλες συναρτήσεις.

1.

Επαναλαμβάνονται οι εντολές επανάληψης for

```
def f1(N):
        sum = 0
        for i in range(N):
                sum = sum + i
        return sum
sum=f1(100) + f1(100) +f1(f1(100))
print sum
```

2.

```
def  product(the_list):
        p = 1
        for  item in the_list:
                p *=item
        return  p
```

3.

```
def readList( ) :
        List = [ ]
        number = input("Δώσε έναν αριθμό ")
        while number !=0 :
                List.append(number)
                number = input("Δώσε έναν
        αριθμό ")
return List
```

```
def avgPositive( List ) :
        s = 0.0
        positives = 0
        for number in List:
                if number > 0 :
                        s = s + number
                        positives = positives + 1
        if positives > 0:
                average = s / positives
        else:
                average = None
        return average

L = readList()
average = avgPositive( L )
print average
```

ΚΕΦΑΛΑΙΟ 6

ΔΙΑΧΕΙΡΙΣΗ ΑΡΧΕΙΩΝ

ΑΣΚΗΣΕΙΣ ΑΡΧΕΙΑ

1)

1) Να ανοίξετε ένα αρχείο με όνομα «file1.txt» για εγγραφή

2) Να δίνετε από το πληκτρολόγιο τις θερμοκρασίες της Αθήνας μέχρι να πληκτρολογήσετε αριθμό μικρότερο από -20.

3) Τις θερμοκρασίες θα τις αποθηκεύετε στο αρχείο σε ξεχωριστές γραμμές.

4) Να κλείσετε το αρχείο.

5) Να ανοίξετε το αρχείο «file1.txt» για διάβασμα.

6) Να διαβάσετε τις θερμοκρασίες και :

Α) να υπολογίσετε και να τυπώσετε τον μέσο όρο των θερμοκρασιών

Β) να υπολογίσετε και να τυπώσετε πόσες θερμοκρασίες είναι μεταξύ 12 με 35 βαθμούς

Γ) Να τυπώσετε την μεγαλύτερη θερμοκρασία

Δ) Να τοποθετήσετε τις θερμοκρασίες σε μια λίστα Α

7) Να κλείσετε το αρχείο.

```
f1=open ("file1.txt", "w")                    //1)
T=input("Dose thermokrasia")                  //2)
while T>-20:
   f1.write(str(T) + "\n")                    //3)
   T=input("Dose thermokrasia")
f1.close()                                    //4)

A=[]
f1=open ("file1.txt", "r")                    //5)
S=0
c1=0
c2=0
max1=-21
for   x   in   f1:                            //6)
   S = S+ int(x)
   c1=c1+1
   if int(x)>=12 and int(x) <=35:
      c2=c2+1
   if int(x)>max1:
      max1=int(x)

   A.append(int(x))

MO=S/c1
print "O Mesos Oros = ", MO
print "Thermokrasies 12-35 = ", c2
print "H megaliteri thermokrasia  =", max1
f1.close()                                    //7)
```

1) Να ανοίξετε και να δημιουργήσετε αρχείο με όνομα Dokimi.txt για εγγραφή (μέσα στον σκληρό δίσκο C).

Στην συνέχεια να γράψετε στο αρχείο τρείς γραμμές όπου κάθε γραμμή θα περιέχει το όνομα, το επώνυμο και την ηλικία σου.

Να κλείσει το αρχείο

2) Να ανοίξετε και το αρχείο με όνομα Dokimi.txt για να προσθέσετε νέα στοιχεία.

Στην συνέχεια να προσθέσετε στο αρχείο τρείς νέες γραμμές όπου κάθε γραμμή θα περιέχει το όνομα, το επώνυμο και την ηλικία ενός συμμαθητή σου.

Να κλείσει το αρχείο

3) Να ανοίξετε και το αρχείο με όνομα Dokimi.txt για διάβασμα.

Στην συνέχεια να τυπώσετε το περιεχόμενο του αρχείου.

Να κλείσει το αρχείο

4) Να δημιουργήσετε μια λίστα A =[45, 22, 53, 7 , 18, 2]

Να ταξινομήσετε την λίστα με την μέθοδο της φυσαλίδας

Στην συνέχεια να ανοίξετε και να δημιουργήσετε αρχείο με όνομα listaB.txt για εγγραφή (μέσα στον σκληρό δίσκο C).

Στην συνέχεια να αποθηκεύσετε στο αρχείο τα στοιχεία της ταξινομημένης λίστας Α. Σε κάθε γραμμή να περιέχετε και ένα στοιχείο της λίστας

Να κλείσει το αρχείο

Να ανοίξετε και το αρχείο με όνομα listaB.txt για διάβασμα.

Στην συνέχεια να τυπώσετε το περιεχόμενο του αρχείου.

Να κλείσει το αρχείο

ΑΣΚΗΣΗ 1	**ΑΣΚΗΣΗ 2**	**ΑΣΚΗΣΗ 3**
f1=open("C:\Dokimi.txt","w") f1.write("onoma \n") f1.write("hlikia \n") f1.write(str(17) + "\n") f1.close()	f1=open("C:\Dokimi.txt","a") f1.write("onoma \n") f1.write("hlikia \n") f1.write(str(16) + "\n") f1.close()	f1=open("C:\Dokimi.txt","r") for x in f1: print x f1.close()

ΑΣΚΗΣΗ 4

```
A=[45, 22, 53, 7 , 18, 2]

N=len(A)

for i in range (1, N,1):
   for j in range(N-1,i-1,-1):
      if A[j]<A[j-1]:
         A[j], A[j-1] = A[j-1],A[j]

f1=open("listaB.txt", "w")

for  x  in A:
   f1.write(str(x) + "\n")

f1.close()

f1=open("listaB.txt", "r")

print f1.read()

f1.close()
```

1) Να γεμίσετε μια ουρά Α με αριθμούς μέχρι να πληκτρολογήσετε το 0.

Στην συνέχεια να ξεχωρίσετε τους θετικούς από τους αρνητικούς αριθμούς. Τους θετικούς να τους τοποθετήσετε στην στοίβα Β και τους αρνητικούς στην στοίβα C.

Να αδειάσετε την ουρά Α.

Την στοίβα C αφού την ταξινομήσετε αύξουσα με ταξινόμηση ευθείας ανταλλαγής (φυσαλίδα) στην συνέχεια να ανοίξετε αρχείο με όνομα arnitikoi.txt και σε κάθε γραμμή να γράψετε κάθε στοιχείο της στοίβας C.

Την στοίβα Β πρώτα να αντιστρέψετε τα στοιχεία της δηλαδή τα πρώτα να γίνουν τελευταία και αντίστροφα π.χ. [18, 3, 15, 7, 19, 20] → [20, 19, 7, 15, 3, 18].

Να τυπωθεί η Β

Να τυπωθεί ο αριθμός των στοιχείων της Β και το τελευταίο στοιχείο της Β

Να τυπωθεί το μεγαλύτερο στοιχείο της Β (χωρίς ταξινόμηση)

Να τυπωθεί πόσα στοιχεία είναι μικρότερα από τον αριθμό 18

Να ανοίξετε αρχείο με όνομα **thetikoi.txt** και σε κάθε γραμμή να γράψετε κάθε στοιχείο της στοίβας Β που είναι μεγαλύτερο από 50. Να κλείσετε το αρχείο.

Να ανοίξετε αρχείο με όνομα **thetikoi.txt** και να προσθέσετε στο τέλος του αρχείου σε κάθε γραμμή κάθε στοιχείο της στοίβας Β που είναι μικρότερο ή ίσο από 50. Να κλείσετε το αρχείο.

Να αδειάσετε την στοίβα Β.

2) Να δίνετε από το πληκτρολόγιο τον αριθμό των πελατών μιας εταιρίας τηλεφωνίας. Κάθε πελάτης χρεώνεται κλιμακωτά ανάλογα με τον χρόνο ομιλίας με βάση τον παρακάτω πίνακα.

Να δίνετε από το πληκτρολόγιο για κάθε πελάτη τον χρόνο ομιλίας και το όνομα του.

Να υπολογιστεί για κάθε πελάτη το ποσό χρημάτων που πρέπει να πληρώσει.

Να αποθηκεύσετε τα ονόματα των πελατών σε μια λίστα Α και τα αντίστοιχα ποσά χρημάτων σε μια λίστα Β.

Να ανοίξετε αρχείο με όνομα **pelates.txt** και να αποθηκεύσετε σε κάθε γραμμή του αρχείου το όνομα του πελάτη, να αφήσετε κενό και δίπλα το ποσό που πρέπει να πληρώσει.

Πάγιο 15€	
Λεπτά ομιλίας	**Χρήματα ανά λεπτό**
1 έως και 3	0,30€
4 έως και 10	0,40€
Πάνω από 10	0,60€

ΑΣΚΗΣΗ 1

```
A=[ ] ; B=[ ]  ; C=[ ]
k=input('Dose arithmo')
while k!=0:

   A.append(k)
   k=input('Dose arithmo')

for x  in A:
   if x>0:
      B.append(x)
   if x<0:
      C.append(x)

while A!=[]:
   A.pop(0)

N=len(C)

for i in range (1, N,1):
   for j in range(N-1,i-1,-1):
      if C[j]<C[j-1]:
         C[j], C[j-1] = C[j-1],C[j]

f1=open("arnitikoi.txt","w")
for x  in C:
   f1.write(str(x) + "\n")
f1.close()
```

```
A= B[:]  ;  B=[ ]
for x in A:
   B.insert(0,x)

print B, len(B), B[-1], max(B)

c=0
for x in B:
   if x<18:
      c+=1
print c

f1=open("thetikoi.txt", "w")

for  x  in B:
   if x > 50:
      f1.write(str(x) + "\n")
f1.close()

f1=open("thetikoi.txt", "a")

for  x  in B:
   if x <= 50:
      f1.write(str(x) + "\n")
f1.close()

while B!=[]:
   B.pop(0)   #gemise me insert
```

ΑΣΚΗΣΗ 2

```
A=[]
B=[]
N=input('Dose arithmo pelaton')

for x in range(N):
   on=raw_input('dose onoma')
   xronos=input('Dose xrono omilias')

   if xronos>=1 and xronos<=3:
      xr=xronos*0.30+15
   if xronos>=4 and xronos<=10:
      xr=3*0.30 + (xronos-3)* 0.40 +15
   if xronos>10:
      xr=3*0.30 + 6* 0.40 + (xronos-10) * 0.60 + 15

   A.append(on)
   B.append(xr)
```

```
f1=open("pelates.txt", "w")

for  x  in range(len(A)):
   f1.write(A[x] + " " + str(B[x]) + "\n")
f1.close()
```

1) Γράψτε την λειτουργία ή το αποτέλεσμα των παρακάτω εντολών:

```
F1= open(“file2.txt”, “w”)

F1.write(“MediaRange \n”)

F1.close()

F1.closed

F1= open(“file2.txt”, “r”)

print F1.mode

print F1.name

F1.seek(3)

print F1.tell()

print F1.read(2)

print F1.tell()

F1.seek(2)

print F1.tell()

F1.seek(2, 1)

print F1.tell()

F1.close()
```

2) Να γεμίσετε τις **λίστες Α και Β** αντίστοιχα με ονόματα και βαθμούς μαθητών μέχρι να πληκτρολογήσετε για όνομα την λέξη **TELOS**.

Στην συνέχεια να αποθηκεύσετε σε αρχείο με όνομα «dataf1.txt» τα ονόματα και τους βαθμούς όσων μαθητών έχουν βαθμό μεγαλύτερο ή ίσο από 18.

Τέλος να διαβάσετε το αρχείο και να τυπώσετε το περιεχόμενο του

3) Να γεμίσετε μια λίστα Α με τα ονόματα των μαθητών της τάξης σας.

Στην συνέχεια διαβάζοντας την λίστα Α να αποθηκεύσετε τα ονόματα σε ένα αρχείο, **αφού πρώτα** τα έχετε ταξινομήσει φθίνουσα με την μέθοδο της ευθείας ανταλλαγής .

Να διαβάσετε το αρχείο και να τυπώσετε το πρώτο όνομα

4) Να δίνετε από το πληκτρολόγιο τον αριθμό των πελατών μιας εταιρείας κινητής τηλεφωνίας.

Για κάθε πελάτη να δίνετε :

Το όνομα του

Τον χρόνο ομιλίας (λεπτά)

Με βάση τον παρακάτω πίνακα να υπολογίζετε για κάθε πελάτη τα χρήματα που θα πληρώσει.

Πάγιο 11€	
Χρόνος Ομιλίας	**Χρήματα ανα λεπτό ομιλίας**
1 έως και 3	0,10€
4 έως και 10	0,12€
Πάνω από 10	0,18€

Τέλος να αποθηκεύετε **για κάθε πελάτη** στο **αρχείο** με όνομα «ΟΤΕ.txt» το όνομα του και το ποσό που πρέπει να πληρώσει

ΑΣΚΗΣΗ 2

```
A=[]
B=[]

x=str(raw_input('Dose onoma'))
while x!="TELOS":
   y=input('Dose vathmo')
   A.append(x)
   B.append(y)
   x=str(raw_input('Dose onoma'))

f1=open("dataf1.txt", "w")

for x in range(len(A)):
   if B[x]>=18:
      Z="O " + A[x] + "exei vathmo = " +
str(B[x])
      f1.write(Z+ " \n")

f1.close()

f1=open("dataf1.txt", "r")

print f1.read()

f1.close()
```

ΑΣΚΗΣΗ 3

```
A=[]

for x in range (11):
   On=str(raw_input('Dose onoma'))
   A.append(On)

N=len(A)
for i in range (1, N,1):
   for j in range (N-1, i-1, -1):
      if A[j] > A[j-1]:
         A[j], A[j-1] = A[j-1], A[j]

f1=open("dataf2.txt", "w")

for x in A:
   f1.write(x + "\n")

f1.close()


f1=open("dataf2.txt", "r")

print f1.readline()

f1.close()
```

ΑΣΚΗΣΗ 4

```
f1=open("OTE.txt", "w")

N=input('Dose arithmo pelaton')

for x  in  range(N):

   On=raw_input('Dose onoma')
   K=input('Dose katanalosi')

   if K>=1 and K<=3:
      Xr= K*0.10+11
   if K>=4 and K<=10:
      Xr= 3*0.10+ (K-3)* 0.12+11
   if K>10:
      Xr= 3*0.10+ 7* 0.12 + (K-10)*0.18 +11

   f1.write(On + "\n")
   f1.write(str(Xr) + "\n")

f1.close()
```

1) Να ανοίξετε ένα αρχείο με όνομα “Text1.txt” και σε κάθε γραμμή να γράψετε τα ονόματα των μαθητών του τμήματος σας (τα οποία θα πληκτρολογήσετε).

2) Να ανοίξετε το αρχείο “Text1.txt” και να τυπώσετε το περιεχόμενο του με τρείς τρόπους.

3) Να ανοίξετε ένα αρχείο με όνομα “Text2.txt” και σε κάθε γραμμή να γράψετε την ηλικία των μαθητών του τμήματος σας (τις οποίες θα πληκτρολογήσετε).

4) Να ανοίξετε το αρχείο “Text1.txt” και να προσθέσετε στο τέλος του το όνομα του διευθυντή του σχολείου

5) Να ανοίξετε το αρχείο “Text2.txt” και να το διαβάσετε ώστε να αποθηκεύσετε το περιεχόμενο του σε ένα άλλο αρχείο με όνομα “Text3.txt” σε αντίστροφη σειρά δηλαδή το τελευταίο πρώτο και το πρώτο τελευταίο.

6) Να ανοίξετε το αρχείο “Text2.txt” και να το διαβάσετε ώστε να τυπώσετε τον μέσο όρο των ηλικιών των μαθητών της τάξης.

7) Να γράψετε ένα πρόγραμμα το οποίο θα ενώνει δύο αρχεία κειμένου σε ένα, τοποθετώντας τα περιεχόμενα του δεύτερου αρχείου μετά από αυτά του πρώτου.

8) Να υλοποιήσετε μια συνάρτηση copy(S, D) η οποία θα δέχεται για είσοδο δύο ονόματα αρχείων και θα δημιουργεί ένα αντίγραφο του αρχείου με όνομα S στο αρχείο με όνομα D.

ΑΣΚΗΣΗ 1

```
f1==open("Text1.txt","w")

for x  in range(20):
   on=raw_input('Dose onoma')
   f1.write(on + "\n")

f1.close()
```

ΑΣΚΗΣΗ 2

1ος

```
f1==open("Text1.txt","r")
for x  in f:
   print x
f1.close()
```

2ος

```
f1==open("Text1.txt","r")
print f1.read()
f1.close()
```

3ος

```
f==open("Text1.txt","r")
for x  in range(20):
   print f1.readline()
f1.close()
```

ΑΣΚΗΣΗ 3

```
f1==open("Text2.txt","w")

for x  in range(20):
   h=input('Dose hlikia')
   f1.write(str(h) + "\n")

f1.close()
```

ΑΣΚΗΣΗ 4

```
f1==open("Text1.txt","a")


on=raw_input('Dose onoma')
f1.write(on + "\n")

f1.close()
```

ΑΣΚΗΣΗ 5

```
A=[]
f1==open("Text2.txt","r")

for x in f1:
   A.insert(0,x)

f1.close()

f2==open("Text3.txt","w")

for x in A:
   f2.write(x + "\n")
f2.close()
```

ΑΣΚΗΣΗ 6

```
f1==open("Text2.txt","r")
S=0.0
c=0
for x  in f1:
   S=S+ int(x)
   c=c+1
f1.close()
print S/c
```

ΑΣΚΗΣΗ 7

```
f1==open("Text1.txt","a")
f2==open("Text2.txt","r")

for x in f2:
   f1.write(x + "\n")

f1.close()
f2.close()
```

ΑΣΚΗΣΗ 8

```
def  copy(S, D):
   f1==open(D,"w")
   f2==open(S,"r")

   for x in f2:
      f1.write(x + "\n")

   f1.close()
   f2.close()
```

ΚΕΦΑΛΑΙΟ 11

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

1) Να δημιουργήσετε μια κλάση **Poli** με ιδιότητες **Onoma και Plithismos**

Να δημιουργήσετε μια μέθοδο με όνομα **Info** που θα τυπώνει το όνομα και τον Πληθυσμό της πόλης

Να δημιουργήσετε τις εξής πόλεις : Αθήνα με πληθυσμό 5000000, Πάτρα με πληθυσμό 800000

Να καλέσετε την μέθοδο info για την πόλη Αθήνα

Να γράψετε κώδικα ώστε να τυπωθεί το όνομα της πόλης με τον μεγαλύτερο πληθυσμό (να ελεγχθεί ποια πόλη έχει τον μεγαλύτερο πληθυσμό).

2) Να δημιουργήσετε μια κλάση **Planites** με ιδιότητες **Onoma, Xroma, Megethos και Apostasi** (από την Γη)

Να δημιουργήσετε μια μέθοδο με όνομα **Thermokrasia** που θα δέχεται ως είσοδο την θερμοκρασία του πλανήτη και θα τυπώνει το όνομα και την θερμοκρασία του πλανήτη.

Να δημιουργήσετε δύο στιγμιότυπα (αντικείμενα πλανήτες) τον Άρη και τον Δία δίνοντας δικές σας ιδιότητες.

Να καλέσετε την μέθοδο **Thermokrasia** για την τον πλανήτη Δία με θερμοκρασία -130 βαθμοί κελσίου.

Να δημιουργήσετε την υποκλάση **Doriforoi** με μέθοδο **info** η οποία θα δέχεται για είσοδο τον πλανήτη στον οποίο είναι δορυφόρος και θα τυπώνει το όνομα, το μέγεθος και τον πλανήτη στον οποίο είναι δορυφόρος.

Να δημιουργήσετε το στιγμιότυπο (αντικείμενο δορυφόρο) την Σελήνη δίνοντας δικές σας ιδιότητες.

Να καλέσετε την μέθοδο **info** για τον δορυφόρο Σελήνη με όνομα πλανήτη, στον οποίο είναι δορυφόρος, την Γη.

ΑΣΚΗΣΗ 1

```
class Poli:
   def __init__(self, Onoma, Plithismos):
      self.Onoma=Onoma
      self.Plithismos=Plithismos

   def info(self):
      print self.Onoma
      print self.Plithismos

Athina=Poli("Athina",5000000)

Patra=Poli("Patra",800000)

Athina.info()

if Athina.Plithismos >Patra.Plithismos :
   print Athina.Onoma

else:
   print Patra.Onoma
```

ΑΣΚΗΣΗ 2

```
class Planites:
   def __init__(self, Onoma, Xroma, Megethos, Apostasi):
      self.Onoma=Onoma
      self.Xroma=Xroma
      self.Megethos=Megethos
      self.Apostasi= Apostasi

   def Thermokrasia(self, T):
      print self.Onoma, T

class Doriforoi(Planites):

   def info(self, P):
      print self.Onoma
      print self.Megethos
      print "Einai doriforos tou ",P.Onoma


Aris=Planites("Aris","Red",1000,500000)
Dias=Planites("Dias","Yellow",10000,800000)
Earth=Planites("Earth","Blue",500,0)

Selini=Doriforoi("Selini","White",100, 5000)

Selini.info(Earth)
```

1) Δημιουργία κλάσης Student με χαρακτηριστικά (attributes):

am (Αριθμός Μητρώου),

name (Ονοματεπώνυμο),

sclass (Τάξη),

avg (Γενικό μέσο όρο βαθμολογίας),

absences (Απουσίες)

Δημιουργία μεθόδων για:

Αύξηση απουσιών (absup)

Προαγωγή στην επόμενη τάξη (taxiup)

Δημιουργούμε στιγμιότυπο ζητώντας με την εντολές input και raw_input τα χαρακτηριστικά.

Γίνονται οι απαραίτητοι έλεγχοι τιμών

Ζητάμε τις απουσίες για τους 9 μήνες του σχ. έτους αυξάνοντας τις απουσίες με τη μέθοδο absup

Αν οι συνολικές απουσίες είναι πάνω από 114 ή ο μέσος όρος βαθμολογίας κάτω από 9.5 ο μαθητής δεν προάγεται αλλιώς προάγεται στην επόμενη τάξη

ΑΣΚΗΣΗ 1

```
class Student:

   def __init__(self, am, name, sclass, avg, absences=0):          #κατασκευαστής της κλάσης
      self.am = am
      self.name = name
      self.sclass = sclass
      self.absences = absences
      self.avg=avg

   def absup(self,amount):                                          #μέθοδος κλάσης
      self.absences+=amount

   def taxiup(self):                                                #μέθοδος κλάσης
      if self.sclass=="A":
         self.sclass="B"
      elif self.sclass=="B":
         self.sclass="Γ"
      elif self.sclass=="Γ":
         self.sclass="Απόφοιτος"

am=int(input('Αριθμός Μητρώου:'))
name=raw_input('Δώσε Οναματεπώνυμο:')
sclass=raw_input('Δώσε τάξη (A,B ή Γ):')
while sclass!='A' and sclass!='B' and sclass!='Γ':
   sclass=raw_input('Δώσε ξανά τάξη (A,B or Γ):')
avg=input("Δώσε μέσο όρο βαθμολογίας (0-20):")
while avg<0 or avg>20:
   avg=input("Δώσε ξανά μέσο όρο βαθμολογίας (0-20):")

student1=Student(am,name,sclass,avg)                    #Δημιουργία στιγμιοτύπου

print student1.am,student1.name,student1.sclass,student1.avg,student1.absences

for i in range (9):
   x=input("Δώσε απουσίες μήνα:")
   student1.absup(x)
   print student1.absences
if student1.absences>114 or student1.avg<9.5:
   print "Δεν πέρασες"
else:
   print "Πέρασες"
   student1.taxiup()
print 'Η τάξη σου είναι:',student1.sclass
```

1) Να δημιουργήσετε μια κλάση Ergazomenoi με ιδιότητες Όνομα και Μισθός

Να δημιουργήσετε μια μέθοδο με όνομα Info που θα τυπώνει το όνομα και τον μισθό του εργαζόμενου

Να δημιουργήσετε μια μέθοδο με όνομα kratiseis που θα δέχεται για είσοδο το ποσοστό των κρατήσεων και θα υπολογίζει και θα επιστρέφει τον καθαρό μισθό του εργαζόμενου δηλαδή τον μισθό χωρίς τις κρατήσεις

Να δώσετε από το πληκτρολόγιο το όνομα και τον μισθό 4 εργαζομένων και στην συνέχεια να δημιουργήσετε 4 αντικείμενα εργαζομένων με τα παραπάνω στοιχεία.

Να τυπώσετε χρησιμοποιώντας την info τα στοιχεία του πρώτου εργαζομένου

Να τυπώσετε χωρίς να χρησιμοποιήσετε την info τα στοιχεία του πρώτου εργαζομένου

Να τυπώσετε τον καθαρό μισθό του τελευταίου εργαζόμενου

Γίνονται 15% κρατήσεις στους μισθούς

ΑΣΚΗΣΗ 1

```
class Ergazomenoi:

   def __init__(self, onoma, misthos):
      self.onoma=onoma
      self.misthos=misthos


   def info(self):
      print self.onoma
      print self.misthos

   def kratiseis(self, K):
      return self.misthos - self.misthos * K / 100

On1=raw_input('Dose onoma 1')
M1=input('Dose mistho 1')
On2=raw_input('Dose onoma 2')
M2=input('Dose mistho 2')
On3=raw_input('Dose onoma 3')
M3=input('Dose mistho 3')
On4=raw_input('Dose onoma 4')
M4=input('Dose mistho 4')

Erg1=Ergazomenoi(On1, M1)
Erg2=Ergazomenoi(On2, M2)
Erg3=Ergazomenoi(On3, M3)
Erg4=Ergazomenoi(On4, M4)

Erg1.info()

print Erg1.onoma
print Erg1.misthos

print Erg4.kratiseis(15)
```

1) Δημιουργήστε τις παρακάτω κλάσεις

Α) Μία *καρέκλα* έχει τις ιδιότητες όνομα, χρώμα, ύψος, υλικό

Β) Μία *γάτα* έχει τις ιδιότητες όνομα, ηλικία, φύλλο αλλά επίσης μπορεί και να εκτελεί διάφορες ενέργειες :

Να φάει (Να τυπώνεται το μήνυμα τι τρώει (είσοδος) και το όνομα της γάτας)

Να κοιμάται (να τυπώνεται το μήνυμα η ΤΑΔΕ γάτα (όνομα γάτας) κοιμάται ΠΟΣΟ (είσοδος, πολύ λίγο, καθόλου)

2) Να κατασκευάσετε μια κλάση με όνομα BOOK και ιδιότητες : Titlos, Sygrafeas, Timi

Δημιουργήστε μια μέθοδο με όνομα Ekptosi η οποία θα έχει για είσοδο το ποσοστό έκπτωσης και θα επιστρέφει την τελική τιμή του βιβλίου με την έκπτωση

Να προσθέσετε την ιδιότητα Selides (αριθμό σελίδων του βιβλίου)

3) Να ορισθεί μια κλάση με ονομα Mathites και με 3 ιδιότητες : Ονομα, Τάξη, και βαθμοί σε 4 μαθήματα (λίστα)

Να φτιάξετε την κλάση που θα περιέχει:

Τον κατασκευαστή

Μια μέθοδο που θα τυπώνει το όνομα και την τάξη του μαθητή

Μια μέθοδο που θα τυπώνει το όνομα του μαθητή και τον μεγαλύτερο βαθμό του

Μια μέθοδο που θα επιστρέφει τον μέσο όρο της βαθμολογίας του

Να δημιουργήσετε 5 αντικείμενα (μαθητές)

ΑΣΚΗΣΗ 1

```
class karekla:

   def __init__(self, onoma, xroma, ypsos, yliko):
      self.onoma=onoma
      self.xroma=xroma
      self.ypsos=ypsos
      self.yliko=yliko


class cat:

   def __init__(self, onoma, hlikia, filo):
      self.onoma=onoma
      self.hlikia=hlikia
      self.filo=filo

   def  eat(self, F):
      print F, self.onoma

   def sleep(self, P):
      print "h ", self.onoma, " koimate ", P
```

ΑΣΚΗΣΗ 2

```
class BOOK:

   def __init__(self, Titlos, Sygrafeas, Timi, Selides ):
      self.Titlos=Titlos
      self.Sygrafeas=Sygrafeas
      self.Timi=Timi
      self.Selides=Selides

   def  Ekptosi (self, P):
      return Timi - Timi * P / 100
```

ΑΣΚΗΣΗ 3

```
class Mathites :

   def __init__(self, onoma, Taksi, B ):
      self.onoma=onoma
      self.Taksi=Taksi
      self.B=B


   def  info (self):
      print self.onoma, self.Taksi

   def max_vathmos(self):
      max1=0
      for  x  in self.B :
         if x > max1:
            max1=x
      print self.onoma, max1

   def Mesos_Oros(self):
      S=0.0
      for  x  in self.B :
         S=S+x

      return S/len(self.B)

M1=Mathites("Petros","A",[18, 19, 15, 16])
M2=Mathites("Maria","A",[19, 19, 17, 15])
M3=Mathites("Nikos","B",[18, 17, 14, 18])
M4=Mathites("Kostas","A",[19, 19, 14, 15])
M5=Mathites("Eleni","B",[18, 17, 20, 16])
```

1) Παράδειγμα δημιουργίας κλάσης Καλαθοσφαιριστή με ιδιότητες όνομα, ύψος, πόντους και φάουλ και μεθόδων σημείωσης πόντων (scores) και φάουλ (makesfoul).

Οι αρχικές τιμές για τους πόντους και τα φάουλ ενός παίκτη (στιγμιότυπο κλάσης Καλαθοσφαιριστή) είναι 0.

Ζητάμε τα ύψη των παικτών h1 και h2

Δημιουργούμε δύο παίκτες (στιγμιότυπα) p1 και p2 με ονόματα Player1 και Player2 και ύψη h1 και h2.

Ακολουθούν οι παρακάτω ενέργειες:

Ο p1 κάνει φάουλ

Ο p2 βάζει 3 πόντους

Ο p1 κάνει φάουλ

Ο p2 βάζει 2 πόντους

Ο p2 κάνει φάουλ

Εμφανίζονται οι πόντοι και τα φάουλ των 2 παικτών

ΑΣΚΗΣΗ 1

```
#Κλαση:Παιχτης Μπασκετ
#Ιδιοτητες-χαρακτηριστικα: ονομα
#Υψος (σε εκατοστα)
#Ποντοι
#αριθμος fouls
#μεθοδοι:
#scores
#mekesfoul
class BasketballPlayer:
    def __init__(self,nm,ht,pt=0,fl=0):
        self.name=nm
        self.height=ht
        self.points=pt
        self.fouls=fl

    def scores(self,amount):
        self.points+=amount
        print self.name,'Σημειωσε ',amount,' ποντους'

    def makesfoul(self):
        self.fouls+=1
        print self.name,'Εκανε φαουλ'

h1=int(input('Δωσε το υψος του α:'))
h2=int(input('Δωσε το υψος του β:'))
p1=BasketballPlayer('Player1',h1)
p2=BasketballPlayer('Player2',h2)
p1.makesfoul()
p2.scores(3)
p1.makesfoul()
p2.scores(2)
p2.makesfoul()
print 'Παικτης1:','Ποντους:',p1.points,'Φαουλς:',p1.fouls
print 'Παικτης2:','Ποντους:',p2.points,'Φαουλς:',p2.fouls
```

1) Να ορίσετε μια κλάση με όνομα Mathitis η οποία θα έχει 4 ιδιότητες :

AM: Αριθμό Μητρώου

Onoma: Επώνυμο

Vathmos: Βαθμολογία τεσσάρων πανελλαδικών μαθημάτων

Taksi : Τάξη (Α, Β ή Γ)

Να ορίσετε τις παρακάτω μεθόδους :

Μια μέθοδο για την εκτύπωση των AM, Onoma, Vathmos, Taksi

Μια μέθοδος για τον υπολογισμό του Μέσου Όρου της βαθμολογίας και εκτύπωσης :

Α) Του Μέσου Όρου

Β) Αν έχει Μέσο όρο μεγαλύτερο ή ίσο από 10 να τυπώνεται «περνάει» αλλιώς να τυπώνεται «δεν περνάει».

2) Να τροποποιήσετε την παραπάνω άσκηση (1) ώστε να δημιουργήσετε μέθοδο που θα γεμίζει την λίστα με τους τέσσερεις βαθμούς από το τέλος

3) Να ορίσετε μια κλάση με όνομα Mathitis η οποία θα έχει 3 ιδιότητες :

Onoma: Επώνυμο

D: Αριθμός δικαιολογημένων απουσιών

A : Αριθμός αδικαιολόγητων απουσιών

Να ορίσετε τις παρακάτω μεθόδους :

Μια μέθοδο για την εκτύπωση των Onoma, D, A

Μια μέθοδος για τον υπολογισμό του συνόλου των απουσιών και εκτύπωσης :

Α) Του συνόλου των απουσιών

Β) Αν έχει σύνολο απουσιών μεγαλύτερο από 114 να τυπώνεται «χάλια» αλλιώς να τυπώνεται «όλα καλά».

ΑΣΚΗΣΗ 1

```
class   Mathitis:


   def  __init__(self, AM, Onoma, Vathmos, Taksi):
      self.AM=AM
      self.Onoma=Onoma
      self.Vathmos=Vathmos[:4]
      self.Taksi=Taksi



   def   Apotelesmata(self):
      print self.AM
      print self.Onoma
      print self.Vathmos
      print self.Taksi

   def   MesosOros(self):
      S=0
      for x in range(len(self.Vathmos)):
         S=S+self.Vathmos[x]
      MO=S/len(self.Vathmos)

      print MO

      if  MO>=10:
         print "pernaei"
      else:
         print "den pernaei"



Maria=Mathitis("1182", "Nikolaou", [16, 12, 11, 18],"G")

Maria.Apotelesmata()

Maria.MesosOros()
```

ΑΣΚΗΣΗ 2

```
class   Mathitis:
    def  __init__(self, AM, Onoma, Vathmos, Taksi):
       self.AM=AM
       self.Onoma=Onoma
       self.Vathmos=Vathmos[:]
       self.Taksi=Taksi



   def   Apotelesmata(self):
       print self.AM
       print self.Onoma
       print self.Vathmos
       print self.Taksi

   def   MesosOros(self):
       S=0
       for x in range(len(self.Vathmos)):
          S=S+self.Vathmos[x]
       MO=S/len(self.Vathmos)

       print MO

       if  MO>=10:
          print "pernaei"
       else:
          print "den pernaei"



   def    Eisagogi(self):
       for x in range(4):
          v=input('Dose Vathmo')
          self.Vathmos.append(v)



Maria=Mathitis("1182", "Nikolaou", [],"G")

Maria.Apotelesmata()

Maria.Eisagogi()

Maria.Apotelesmata()
Maria.MesosOros()
```

ΑΣΚΗΣΗ 3

```
class    Mathitis:

   def  __init__(self, Onoma, D, A):
      self.Onoma=Onoma
      self.D=D
      self.A=A


   def    Apotelesmata(self):
      print self.Onoma
      print self.D
      print self.A

   def   Apousies(self):
      S=self.A + self.D
      print S

      if  S>114:
         print "xalia"
      else:
         print "ola kala"


Maria=Mathitis("Nikolaou", 118, 56)

Maria.Apotelesmata()

Maria.Apousies()
```

ΑΣΚΗΣΕΙΣ

ΓΕΝΙΚΕΣ

ΑΣΚΗΣΕΙΣ ΓΕΝΙΚΕΣ

1) Να γραφεί πρόγραμμα σε Python το οποίο θα διαβάζει, με κατάλληλη συνάρτηση της Python και να τυπώνει τυχαίους αριθμούς συνεχώς μέχρις ότου το άθροισμα τους να ξεπεράσει το 50 ή μόλις 10 αριθμοί (δηλ. το πλήθος των αριθμών που διαβάστηκαν να είναι 10).

Οι αριθμοί που θα διαβάζονται θα πρέπει να είναι μεταξύ 5 και 15

2) Να γραφεί πρόγραμμα σε Python το οποίο θα διαβάζει, με κατάλληλη συνάρτηση της Python και να τυπώνει τυχαίους αριθμούς συνεχώς μέχρις ότου συμπληρώσει 10 αρνητικούς και περιττούς αριθμούς.

Οι αριθμοί που θα διαβάζονται θα πρέπει να είναι μεταξύ -100 και 100

3) Να γραφεί πρόγραμμα σε Python το οποίο θα εκτυπώνει την προπαίδεια από το 1 έως και το 9. (Να χρησιμοποιήσετε μόνο while)

4) Η Εταιρεία ύδρευσης της περιοχής μας ΔΕΥΑΠ χρεώνει κλιμακωτά τους πελάτες της ανάλογα με τα κυβικά μέτρα νερού που κατανάλωσαν, σύμφωνα με τον παρακάτω πίνακα:

Κυβικά μέτρα νερού	**Τιμή ανά κυβικό μέτρο**
Από 0 έως και 400	25 €
Από 401 έως και 900	30 €
Πάνω από 900	36 €

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Να περιέχει συνάρτηση **vres_kostos** η οποία να δέχεται σαν είσοδο τα κυβικά που κατανάλωσε ο πελάτης και να επιστρέφει το κόστος νερού σύμφωνα με τον παραπάνω πίνακα και με ***κλιμακωτή χρέωση***.

Μονάδες 5

Γ2.Να διαβάζει το Επώνυμο και το όνομα του καταναλωτή και τα κυβικά που κατανάλωσε και να τα καταχωρεί στις λίστες **EP, ON, KYB** αντίστοιχα.

Μονάδες 5

Γ3. Η διαδικασία να επαναλαμβάνεται μέχρι να δοθεί για επώνυμο η λέξη "**TELOS**".

Μονάδες 2

Γ4. Με τη βοήθεια της συνάρτησης του Γ1 να υπολογίζει και να εμφανίζει το κόστος νερού για κάθε ένα πελάτη. Να το καταχωρεί σε μία νέα λίστα **KOSTOSN**.

Μονάδες 4

Γ5. Με τη βοήθεια τις **συνάρτησης bubbleSort** για 4 λίστες να ταξινομήσετε τις λίστες ως προς το κόστος νερού με αύξουσα ταξινόμηση και να εμφανίσετε τα ονοματεπώνυμα των 3 πελατών με το μεγαλύτερο κόστος νερού καθώς επίσης και τα κυβικά μέτρα νερού που κατανάλωσαν. (Θεωρούμε ότι μας δόθηκαν τουλάχιστον 3 πελάτες)

Μονάδες 6

Γ6. Για κάθε πελάτη υπάρχει ένας επιπλέον φόρος που είναι για τη δημιουργία έργων ύδρευσης και αποχέτευσης. Ο φόρος αυτός είναι το **80%** του κόστους του νερού. Να υπολογίσετε και να εμφανίσετε το φόρο έργων και το τελικό ποσό που πρέπει να πληρώσει ο κάθε πελάτης.

Μονάδες 3

5) Στους αγώνες άλματος με σκι ο κάθε αθλητής κάνει δύο άλματα. Για κάθε άλμα παίρνει μία βαθμολογία. Η τελική βαθμολογία προκύπτει από το άθροισμα των δύο βαθμολογιών. Ο νικητής του αγωνίσματος είναι αυτός που θα έχει τη μεγαλύτερη τελική βαθμολογία.

Να γράψετε πρόγραμμα σε Python το οποίο:

Δ1. Να διαβάζει το επώνυμο, το όνομα, τη βαθμολογία στο πρώτο άλμα, τη βαθμολογία στο δεύτερο άλμα και τη χώρα του αθλητή.

Μονάδες 4

Δ2. Η παραπάνω διαδικασία να επαναλαμβάνεται για 80 αθλητές.

Μονάδες 2

Δ3. Να καταχωρίζει τα στοιχεία στις λίστες EP,ON,ALMA1, ALMA2 και XWRA αντίστοιχα.

Μονάδες 4

Δ4. Για κάθε αθλητή να υπολογίζει και να εμφανίζει τη συνολική βαθμολογία του (αθροίζοντας τα δύο άλματα που έκανε) και να την καταχωρεί σε μία νέα λίστα SYN.

Μονάδες 4

Δ5. Να υπολογίζει και να εμφανίζει το άθροισμα της συνολικής βαθμολογίας όλων των αθλητών από τη ΓΕΡΜΑΝΙΑ

Μονάδες 4

Δ6. Να υπολογίζει και να εμφανίζει το επώνυμο και το όνομα του αθλητή με τη μεγαλύτερη συνολική επίδοση

Μονάδες 4

Δ7. Να υπολογίζει και να εμφανίζει το πλήθος των αθλητών που είχαν μεγαλύτερη βαθμολογία στο δεύτερο άλμα σε σχέση με το πρώτο άλμα.

Μονάδες 3

ΑΣΚΗΣΗ 1	**ΑΣΚΗΣΗ 2**

import random
 	
x= random.randint(5, 15)
print x
S=0
c=0
while S+x <=50 and c<=10:
  S=S+x
  c=c+1
  print ".....",x
  x= random.randint(5, 15)
  print x
	import random
	c=1
	while c<=10:
	  x= random.randint(-100, 100)
	  if x%2==1 and x<0:
	  c=c+1
	  print x

ΑΣΚΗΣΗ 3	**ΑΣΚΗΣΗ 3**

for x in range(1, 10):
  print "To ", x
  for y in range (1, 10):
  print x,"*",y,"=", x*y
	x=1
	while x <=9:
	  print "To ", x
	  y=1
	  while y <=10:
	  print x,"*",y,"=", x*y
	  y=y+1
	  x=x+1

ΑΣΚΗΣΗ 4

```
def  vres_kostos(Kivika):
   if Kivika >=0 and Kivika <=400:
      Xr=Kivika*25
   elif Kivika >=401 and Kivika <=900:
      xr=400*25 + (Kivika-400)*30
   else:
      xr=400*25 + 500*30 + (Kivika-
900)*36


   return Xr


def bubbleSort(A,B,C,D):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1, i-1,-1):
         if D[j]<D[j-1]:
            A[j],A[j-1]=A[j-1], A[j]
            B[j],B[j-1]=B[j-1], B[j]
            C[j],C[j-1]=C[j-1], C[j]
            D[j],D[j-1]=D[j-1], D[j]
EP=[]
ON=[]
KYB=[]
KOSTOSN=[]


ep=str(raw_input('Dose eponimo'))
while  ep!="TELOS":
   on=str(raw_input('Dose onoma'))
   K=input('Dose kivika')
   EP.append(ep)
   ON.append(on)
   KYB.append(KYB)
   ep=str(raw_input('Dose eponimo'))
```

```
for x in range(len(KYB)):
   KOSTOSN.append(vres_kostos(KYB[x]))
   print vres_kostos(KYB[x])


bubbleSort(EP,ON,KYB,KOSTOSN)


for x in range(-1, -4,-1):
   print EP[x], ON[x], KYB[x]


for x in range(KOSTOSN):
   Foros=KOSTOSN[x]*80/100
   pliromi=KOSTOSN[x] + Foros
   print Foros, pliromi
```

ΑΣΚΗΣΗ 5

```
EP=[] ; ON=[] ; ALMA1=[] ; ALMA2=[] ; XWRA=[] ; SYN=[]
for x in range(80):
   ep=str(raw_input('Dose eponimo'))
   on=str(raw_input('Dose onoma'))
   A1=input('Dose to Alma 1')
   A2=input('Dose to Alma 2')
   xwra=str(raw_input('Dose xora'))
   EP.append(ep)
   ON.append(on)
   ALMA1.append(A1)
   ALMA2.append(A2)
   XWRA.append(xwra)

for x in range(80):
   print "Synoliki Vathmologia tou ",EP[x], " einai ", ALMA1[x]+ALMA2[x]
   SYN.append(ALMA1[x]+ALMA2[x])

S=0
for x in range(80):
   if XWRA[x]=="Germania":
      S= S+ SYN[x]
max1=0
for x in range(80):
   if SYN[x]>max1:
      max1=SYN[x]
      Z=x

print "megaliteri vathmologia exei o ", ON[Z], EP[Z]

c=0
for x in range(80):
   if ALMA2[x]>ALMA1[x] :
      c=c+1
print c
```

ΓΕΝΙΚΕΣ ΑΣΚΗΣΕΙΣ

1) Η πενταμελής επιτροπή ενός σχολικού συγκροτήματος θα οργανώσει μια εκπαιδευτική επταήμερη εκδρομή στην Ιταλία. Κάποιο ξενοδοχείο τους πρότεινε την ημερήσια προσφορά που παρουσιάζεται στον παρακάτω πίνακα και αφορά τους μαθητές που θα συμμετάσχουν (συμπεριλαμβανομένης της πενταμελούς επιτροπής) και τους συνόδους καθηγητές.

Αριθμός Μαθητών	**Κόστος ανά άτομο ημερησίως**	**Ποσοστό έκπτωσης σε καθηγητές και πενταμελή επιτροπή**
Μέχρι 50 μαθητές	30€ το άτομο	10%
Μέχρι 80 μαθητές	25€ το άτομο	15%
Περισσότεροι από 80 μαθητές	20€ το άτομο	25%

Ο αριθμός των καθηγητών που πρέπει να συνοδέψουν τους μαθητές στην εκδρομή εξαρτάται από το πλήθος των μαθητών. Έτσι μέχρι 50 μαθητές συνοδεύονται από 4 καθηγητές, μέχρι 80 μαθητές συνοδεύονται από 7 καθηγητές, ενώ απαιτούνται 10 καθηγητές αν οι μαθητές υπερβαίνουν τους 80.

Να γράψετε αλγόριθμο ο οποίος:

Α. Θα δέχεται τον αριθμό των μαθητών που θα συμμετάσχουν στην εκδρομή.

Β. Θα εμφανίζει το κόστος διαμονής για έναν μαθητή και για το σύνολο της πενταμελούς επιτροπής και για τις επτά (7) ημέρες της εκδρομής.

Γ. Να εμφανίζει ένα μήνυμα που θα περιλαμβάνει το πλήθος των ατόμων (καθηγητές και μαθητές) που συμμετέχουν και το συνολικό κόστος της επταήμερης εκδρομής, της μορφής: «Στην εκδρομή συμμετέχουν ... άτομα και το συνολικό κόστος είναι … ευρώ».

2) Ο ΕΝΦΙΑ υπολογίζεται με βάση τα τετραγωνικά μέτρα ενός ακινήτου. Εφαρμόζεται κλιμακωτή χρέωση σύμφωνα με τον επόμενο πίνακα:

Τετραγωνικά Μέτρα Ακινήτου	**Χρέωση ανά τετραγωνικό μέτρο**
Από 0 εώς και 80	20€
Από 81 εώς και 150	40€
Από 151 εώς και 300	100€
Από 301 και άνω	250€

Στο ποσό που προκύπτει από την χρέωση υπολογίζεται ο ΦΠΑ με συντελεστή 23%. Το τελικό ποσό προκύπτει από την άθροιση της χρέωσης και του ΦΠΑ. Να γράψετε πρόγραμμα το οποίο:

Α) Θα διαβάζει τα τετραγωνικά μέτρα του ακινήτου.

Β) Θα υπολογίζει χρέωση του ΕΝΦΙΑ σύμφωνα με την παραπάνω τιμολογιακή πολιτική.

Γ) Θα υπολογίζει και θα εκτυπώνει τον ΦΠΑ.

Δ) Θα υπολογίζει και θα εκτυπώνει το τελικό ποσό με κατάλληλο μήνυμα.

ΑΣΚΗΣΗ 1

```
N=input('Δωσε αριθμό μαθητών')

if N>=1 and N< 50:
   K=4
   MK=30*7
   SK=(30-30*10/100)*K*7
if N>=50 and N<80:
   K=7
   MK=25*7
   SK=(25-25*15/100)*K*7
if N>=80 :
   K=10
   MK=20*7
   SK=(20-20*25/100)*K*7

print "κόστος διαμονής για έναν μαθητή ",MK

print "κόστος διαμονής για την επιτροπή ",Sk

print "Στην εκδρομή συμμετέχουν", N+K ,"άτομα και το συνολικό κόστος είναι ", MK+SK , "ευρώ"
```

ΑΣΚΗΣΗ 2

```
N=input('Δωσε τετραγωνικά')

if N>=0 and N<=80:
   xr=N*20
if N>=81 and N<=150:
   xr= 80*20+(N-80)*40
if N>=151 and N<=300:
   xr= 80*20+70*40 + (N-150)*100
if N>300 :
   xr=80*20+70*40 + 150*100 + (N-300)*250

print xr

print "ΦΠΑ ",xr *23/100

print "τελικό ποσό  ", xr + xr *23/100
```

1) Η εταιρεία DeltaTime Systems εξοπλίζει αθλητικούς αγώνες με συστήματα χρονομέτρησης. Τα συστήματα αυτά δημιουργούν δυο παράλληλες λίστες: την λίστα ON με το όνομα κάθε αθλητή και την λίστα XR με τον χρόνο που χρειάστηκε ο αθλητής να τερματίσει.

Να αναπτύξετε αλγόριθμο που, με δεδομένες τις παραπάνω λίστες:

Α) Να δίνετε το όνομα και τον χρόνο κάθε αθλητή μέχρι να πληκτρολογήσετε όνομα αθλητή την λέξη «ΤΕΛΟΣ» ή την λέξη «τελος»

Β) Στην συνέχεια θα δημιουργεί την λίστα Τ που είναι παράλληλη με τις άλλες λίστες και περιέχει τη θέση που κατέλαβε κάθε αθλητής.

Γ) Θα εκτυπώνει αλφαβητικά τα ονόματα των αθλητών και τη θέση που κατέλαβαν.

2) Τρία σχολεία της Αθήνας συμμετέχουν σε μαθητικό διαγωνισμό. Από κάθε σχολείο αγωνίζονται 30 μαθητές. Δίνονται για κάθε σχολείο δυο παράλληλες λίστες με τα όνομα και το χρόνο κάθε μαθητή, με τους χρόνους σε αύξουσα διάταξη. Θεωρούμε ότι οι χρόνοι όλων των μαθητών είναι διαφορετικοί μεταξύ τους.

Να αναπτύξετε αλγόριθμο ο οποίος θα διαβάζει το όνομα ενός μαθητή του τρίτου σχολείου και θα εκτυπώνει πόσοι μαθητές είχαν χρόνο μικρότερο από αυτόν, **από τα άλλα δύο σχολεία**, καθώς και τα ονόματά τους ξεκινώντας από αυτόν που τερμάτισε πρώτος.

3) Η Γ λυκείου διοργανώνει λαχειοφόρο αγορά ώστε να συγκεντρώσει χρήματα για εκδρομή στην Ρόδο. Όλοι οι λαχνοί είναι αριθμημένοι με τετραψήφιο αριθμό και πουλήθηκαν όλα.

Τα δώρα θα μοιραστούν ως εξής:

Όποιοι έχουν λαχνό με αριθμό που το τελευταίο ψηφίο είναι ίδιο με αυτό του τυχερού λαχνού κερδίζουν μια μπλούζα.

Όποιοι έχουν λαχνό με αριθμό που τα 2 πρώτα ή 2 τελευταία ψηφία είναι ίδια με αυτά του τυχερού λαχνού κερδίζουν μια δωροεπιταγή.

Όποιοι έχουν λαχνό με αριθμό που τα 3 τελευταία ψηφία είναι ίδια με αυτά του τυχερού λαχνού κερδίζουν ένα mp3 player.

Όποιος έχει τον τυχερό αριθμό κερδίζει μια συσκευή κινητού.

Να αναπτύξετε αλγόριθμο που με δεδομένες λίστες ON, CODE που περιέχουν τα ονόματα και τους αριθμούς των λαχνών που κατέχουν.

Το πρόγραμμα θα διαβάζει τον τυχερό αριθμό που κληρώθηκε και θα εκτυπώνει τα ονόματα των τυχερών ακολουθούμενα από τα δώρα που κερδίζουν.

ΑΣΚΗΣΗ 1

```
ON=[]
XR=[]
T=[]

on=str(raw_input("Δωσε ονομα"))

while on!="TELOS"  and  on!= "telos":
   xr=input("Δώσε χρόνο")

   ON.append(on)
   XR.append(xr)

   on=str(raw_input("Δωσε ονομα"))

N=len(XR)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if XR[j]<Xr[j-1]:
         XR[j], XR[j-1]= XR[j], XR[j-1]
         ON[j], ON[j-1]= ON[j], ON[j-1]

T=range(1,len(XR)+1)

N=len(XR)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if ON[j]<ON[j-1]:
         XR[j], XR[j-1]= XR[j], XR[j-1]
         ON[j], ON[j-1]= ON[j], ON[j-1]
         T[j], T[j-1]= T[j], T[j-1]

for x in range(len(ON)):
   print ON[x], T[x]
```

ΑΣΚΗΣΗ 2

```
L = [ ]
XRL=[]

while   XRA != [ ] and   XRB != [ ] :
   if   XRA[0] < XRB[0] :
      XRL.append( XRA.pop(0) )
      L.append( A.pop(0) )
   else:
      XRL.append( XRB.pop(0) )
      L.append( B.pop(0) )

XRL=XRL + XRA + XRB
L=L + A + B


on=raw_input("Δωσε όνομα")

for x in range(len(C)):
   if C[x]==on:
      xronos=XRC[x]

for x in range(len(XRL)):
   if XRL[x] < xronos:
      print L[x]
```

ΑΣΚΗΣΗ 3

```
L=raw_input("Δωσε τυχερό αριθμό")

for x  in range(len(CODE)):
   Y=CODE[x]

   if Y[-1]== L[-1]:
      print "O ", ON[x], " κερδίζει μια μπλούζα"

   if Y[ : 2]== L[ : 2] or Y[ 2 : ]== L[ 2 : ]:
      print "O ", ON[x], " κερδίζει μια δωροεπιταγή"

   if  Y[ 1 : ]== L[ 1 : ]:
      print "O ", ON[x], " κερδίζει ένα mp3 player"

   if  Y == L:
      print "O ", ON[x], " κερδίζει μια συσκευή κινητού"
```

1) Διαθέτουμε μια λίστα ΟΝ[10.000.000] με όλα τα ονόματα των ελλήνων. Να αναπτύξετε αλγόριθμο που:

Α. θα δημιουργεί λίστα Μ με όλα τα διαφορετικά ονόματα που υπάρχουν.

Β. Θα εκτυπώνει κάθε μοναδικό όνομα καθώς και το πόσες φορές συναντάται. Ποιο είναι το πιο δημοφιλές όνομα;

2) Ένας εκδοτικός οίκος χρησιμοποιεί 35 διανομείς για τη διακίνηση των βιβλίων του. Στο τέλος κάθε μήνα καταγράφονται οι πωλήσεις που πραγματοποιήθηκαν από κάθε διανομέα ώστε να υπολογιστεί και το μπόνους που θα του αποδοθεί. Είναι ευνόητο ότι οι πωλήσεις ενός διανομέα σε χρονικό διάστημα ενός μηνός δεν είναι κατ' ανάγκην 30. Το ποσό του μπόνους υπολογίζεται κλιμακωτά ανάλογα με το ποσό των μηνιαίων πωλήσεων κάθε διανομέα σύμφωνα με τον παρακάτω πίνακα:

Συνολικές μηνιαίες πωλήσεις διανομέα (€)	**Μπόνους %**
Μέχρι και 200	0
Άνω των 200 μέχρι και 1000	1.5
Άνω των 1000	4

Να αναπτύξετε αλγόριθμο που θα επιτελεί τις παρακάτω ενέργειες:

Α) Για κάθε διανομέα:

i. θα διαβάζει το όνομά του και θα το καταχωρεί σε λίστα ΟΝ καθώς και το μηνιαίο βασικό μισθό του και θα το καταχωρεί σε λίστα ΒΜ.

ii. θα διαβάζει επαναληπτικά τα ποσά των πωλήσεων που πέτυχε τον προηγούμενο μήνα και θα υπολογίζει τις συνολικές μηνιαίες πωλήσεις. Η επαναληπτική διαδικασία θα ολοκληρώνεται όταν εισαχθεί αρνητικός αριθμός ή το μηδέν.

iii. θα υπολογίζει το μπόνους που θα λάβει και θα το εκτυπώνει.

Β) Θα δημιουργεί λίστα Τ, που θα περιέχει τις τελικές μηνιαίες απολαβές κάθε διανομέα.

Γ) Να εκτυπώνονται τα ονόματα όσων διανομέων είχαν το δεύτερο μεγαλύτερο τελικό μισθό μεταξύ των υπαλλήλων του εκδοτικού οίκου.

Δ) Θα ελέγχει ποιος διανομέας έχει τον υψηλότερο τελικό μισθό που να είναι ταυτόχρονα μικρότερος από 600 € και θα εκτυπώνει το όνομά του. Αν δεν υπάρχει τέτοιος, να εκτυπώνεται κατάλληλο μήνυμα.

ΑΣΚΗΣΗ 1

```
ON=["Marios", "Kostas", "Petros", "Marios", "Nikos", "Marios", "Petros", "Nikos", "Marios", "Petros"]


ON2=ON[:]
M=[]
T=[]

c2=0
while ON2!=[]:
   c=0
   for x in ON2:
      if x==ON2[0]:
         c=c+1
   M.append(ON2[0])
   T.append(c)

   c=0
   while M[c2]  in ON2:
      if ON2[c] == M[c2]:
         ON2.pop(c)
      else:
         c=c+1
   c2+=1

max1=0

for x in range(len(M)):
   print M[x], T[x]
   if T[x]>max1:
      max1=T[x]
      Z=M[x]

print "το πιο δημοφιλές όνομα", Z
```

ΑΣΚΗΣΗ 2

```
ON=[]
BM=[]
T=[]
for x  in range(35):
   S=0
   on=str(raw_input("Δωσε ονομα"))
   bm=input("Δωσε βασικο μισθό")

   p=input("Δωσε αριθμό πωλήσεων")
   while p >0:
      S=S+p
      p=input("Δωσε αριθμό πωλήσεων")

   if S>=0 and S<=200:
      B=S*0
   if S>=201 and S<=1000:
      B=200*0 + (S-200)*1.5/100
   if S>1000:
      B=200*0 + 800*1.5/100 + (S-1000)* 4/100

   print "Το μπόνους είναι", B
   KM=bm+B
   T.append(KM)

max1=0
for x in T:
   if x > max1:
      max1=x
c=0
for x  in range(len(T)):
   if T[x]==max1:
      print ON[x]
      c=c+1

if max1>600:
   print "Έχουν μισθό μεγαλύτερο απο 600",c
else:
   print "Δεν υπάρχει μισθος μεγαλύτερος απο 600"
```

1) Τον χειμώνα του 2025 , καταστροφικές πλημμύρες έπληξαν το νησί της Κρήτης, αφήνοντας πίσω τους ανυπολόγιστες ζημιές. Η περιφέρεια Κρήτης, ανακοίνωσε κονδύλι ύψους **10.000.000** ευρώ με σκοπό την κάλυψη κάποιων ζημιών σε δημόσιους χώρους. Οι ενδιαφερόμενες εταιρίες θα κάνουν την προσφορά τους, ανακοινώνοντας και την περιοχή που θα γίνει η αντίστοιχη εργασία για αποκατάσταση ζημιάς.

Η περιφέρεια ελέγχει το έργο και εφόσον επαρκούν τα χρήματα του κονδυλίου, δέχεται το έργο και προχωρά στην αποταμίευση του ποσού. Σε περίπτωση που δεν επαρκεί το υπόλοιπο κονδύλι της, απορρίπτει την προσφορά της εταιρίας.

Να γραφεί πρόγραμμα το οποίο:

Γ1. Για κάθε εταιρεία που θέλει να κάνει προσφορά να διαβάζει το **ύψος της προσφοράς** και την **περιοχή** που θα γίνει η αντίστοιχη εργασία. Στη συνέχεια εφόσον υπάρχει επάρκεια στο κονδύλι η περιφέρεια αποδέχεται την προσφορά, διαφορετικά απορρίπτεται.

Γ2. Το πρόγραμμα συνεχίζει τις παραπάνω ενέργειες μέχρι να τελειώσουν τα χρήματα τις Περιφέρειας ή απορριφθούν **τρεις συνεχόμενες** προσφορές.

Γ3. Να υπολογίζει και να εμφανίζει ανάμεσα σε όλες τις προσφορές που έγιναν, το ποσοστό των προσφορών που έγιναν αποδεκτές από την περιφέρεια.

Γ4. Τέλος να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα

i. τα ονόματα των περιοχών με τις δύο μικρότερες προσφορές που έγιναν αποδεκτές

ii. τον μέσο όρο των προσφορών που δεν έγιναν αποδεκτές εφόσον υπήρχαν τέτοιες.

Σημείωση: Θεωρούμε ότι τουλάχιστον 2 προσφορές έγιναν αποδεκτές

2) Μια εισαγωγική εταιρεία **εισήγαγε 10000 οθόνες προς 40€** τη μία και τις πουλά σε τιμές χονδρικής, χρεώνοντας κάθε οθόνη ανάλογα με το μέγεθος της παραγγελίας κλιμακωτά, όπως φαίνεται στον παρακάτω πίνακα:

Αριθμός οθονών	**Τιμή ανά οθόνη (σε €)**
1 έως και 20	80
21 έως και 50	70
51 έως και 90	60
Πάνω από 90	55

Να δημιουργήσετε πρόγραμμα σε Python το οποίο:

Γ1. Για κάθε πελάτη:

α) Να διαβάζει το **όνομα** του και τον **αριθμό των οθονών** που επιθυμεί να αγοράσει.

β) Να ελέγχει αν η εταιρεία διαθέτει επαρκές υπόλοιπο οθονών προς πώληση.

γ) Αν υπάρχει επαρκές υπόλοιπο να υπολογίζει και να εμφανίζει το κόστος της παραγγελίας. Σε αντίθετη περίπτωση να ενημερώνει τον πελάτη με κατάλληλο μήνυμα ότι το υπόλοιπο δεν επαρκεί για την αγορά που επιθυμεί.

Γ3. Να επαναλαμβάνει τη διαδικασία, μέχρι να εισαχθεί ως όνομα πελάτη η λέξη ‘’ΤΕΛΟΣ’’ ή να μην υπάρχει επαρκές υπόλοιπο για την κάλυψη της παραγγελίας.

Γ4. Στο τέλος να υπολογίζει και να εμφανίζει:

α) συνολικά τα κέρδη της εταιρείας.

β) Το μέσο όρο αριθμό οθονών ανά πελάτη (να είναι ακέραιος αριθμός).

γ) Το ποσοστό των πελατών που αγόρασαν πάνω από 50 οθόνες.

δ) Το όνομα του πελάτη με το μεγαλύτερο κόστος παραγγελίας.

Σημείωση: Δεν απαιτείται κανένας έλεγχος για τα δεδομένα εισόδου.

3) Στην Αθήνα τα λεωφορεία έχουν χωρητικότητα 40 επιβατών. Ένα λεωφορείο που εκτελεί την διαδρομή Άλιμος – Ελληνικό κάνει 10 στάσεις.

Να δημιουργήσετε 4 λίστες :

Την Α όπου θα καταχωρούνται οι επιβάτες που εισέρχονται στο λεωφορείο σε κάθε στάση

Την Β όπου θα καταχωρούνται οι επιβάτες που εξέρχονται από το λεωφορείο σε κάθε στάση

Την C όπου θα καταχωρούνται το σύνολο των επιβατών μέσα στο λεωφορείο σε κάθε στάση

Την D όπου θα καταχωρείτε το όνομα της κάθε στάση

Α) Να βρεθεί και να τυπωθεί το όνομα της στάσης όπου το λεωφορείο είχε τους περισσότερους επιβάτες

Β) Αν υπάρχει στάση όπου μπήκαν και βγήκαν οι περισσότεροι επιβάτες τότε να τυπωθεί το όνομα της αλλιώς να τυπωθεί ανάλογο μήνυμα.

Γ) Να δώσετε το όνομα μιας στάσης και αν αυτή υπάρχει να τυπωθεί ο αριθμός των εισερχομένων και εξερχομένων επιβατών αλλιώς να τυπωθεί ανάλογο μήνυμα (Να χρησιμοποιηθεί η συνάρτηση της δυαδικής αναζήτησης - binarySearch)

ΑΣΚΗΣΗ 1

```
K=10000000
pr=0
A=[]
B=[]
d=0
MA=0.0
P=input('Dose prosfora')
while P>K:
      pr=pr+1
      d=d+1
      MA=MA+P
      P=input('Dose sosti prosfora')

while K>=P and pr<=3 :
   pr=0
   On=str(raw_input('Dose perioxi'))
   A.append(On)
   B.append(P)
   K=K-P
   P=input('Dose prosfora')
   while P>K:
      pr=pr+1
      d=d+1
      MA=MA+P
      P=input('Dose sosti prosfora')

S=len(A)+d

print "pososto apodekton prosforon =", 100*len(A)/S

N=len(B)
for  i in range(1,N,1):
   for  j in range(N-1,i-1,-1):
      if B[j]<B[j-1]:
         B[j], B[j-1]=B[j-1],B[j]
         A[j], A[j-1]=A[j-1],A[j]

print A[0], A[1]

print "Mesos oros mi apodekton prosforon=", MA/d
```

ΑΣΚΗΣΗ 2

```
C=10000
A=[]  ;  B=[]  ;  K=[]
On=str(raw_input('Dose onoma pelati'))
P=input('Dose paragelia')
if P>C :
   print "Den yparxei eparkes ypoloipo"

while On!="TELOS" and P<=C :
   A.append(On)
   B.append(P)
   if P>=1 and P<=20:
      Xr=P*80
   if P>=21 and P<=50:
      Xr=20*80 + (P-20)*70
   if P>=51 and P<=90:
      Xr=20*80 + 30*70+ (P-50)*60
   if P>=91:
      Xr=20*80 + 30*70+ 40*60+ (P-90)*55

   print Xr
   K.append(Xr)
   C=C-P
   On=str(raw_input('Dose onoma pelati'))
   P=input('Dose paragelia')
   if P>C :
      print "Den yparxei eparkes ypoloipo"

S=0
for x  in  K:
   S=S+x
print  "Kerdi = ", S-10000*40

S=0
z=0
for x  in  B:
   S=S+x
   if x  > 50:
      z=z+1

print "mesos oros othonon = ", S/len(B)
print "Pososto me pano apo 50 othones =", 100*z/len(B)

max1=0
for  x  in range (len(K)):
   if K[x] >max1:
      max1=K[x]
      onoma=A[x]
print onoma
```

ΑΣΚΗΣΗ 3

```
def    binarysearch(ON, ONOMA) :
   first = 0
   last = len(ON)-1
   found = False
   while   found== False :

      mid = ( first + last ) // 2
      if   ON[ mid ] == ONOMA :
         found = True
      elif  ON[ mid ] < ONOMA :
         first = mid + 1
      else:
         last = mid-1

   if  found == True:
       return  mid
   else:
      return  -1

A=[]
B=[]
C=[]
D=[]
S=0
for x  in range(10):
   On=str(raw_input('Dose onoma stasis'))
   D.append(On)
   Eis=input('Dose aritmo eiserxomenon')
   Ex=input('Dose aritmo exerxomenon')

   if  S + Eis – Ex > 40:         # Έλεγχος ώστε να μην έχει το λεωφορείο περισσότερους από 40
       Eis = 40 – S – Ex        # θα αφήσει εκτός λεωφορείου επιβάτες (όσους δεν χωράνε)
       S=40
   else:
       S= S + Eis – Ex

   A.append(Eis)
   B.append(Ex)
   C.append(S)

#a
max1=0
for x  in range(10):
   if C[x]> max1:
      max1 = C[x]
      stasi=D[x]
print "Stasi me tous perisoterous epivates ", stasi
```

```
#b
maxEis=0
maxEx=0
for x  in range(10):
   if A[x]>maxEis:
      maxEis=A[x]
      TEis=D[x]
   if B[x]>maxEx:
      maxEx=B[x]
      TEx=D[x]
if TEis==TEx:
   print  TEx
else:
   print "Den yparxei"

#c
On_Stasis=raw_input('Dose onoma stasis')

N=len(D)
for  i in range(1,N,1):
   for  j in range(N-1,i-1,-1):
      if D[j]<D[j-1]:
         D[j], D[j-1]=D[j-1],D[j]
         B[j], B[j-1]=B[j-1],B[j]
         A[j], A[j-1]=A[j-1],A[j]
         C[j], C[j-1]=C[j-1],C[j]

Y=binarySearch(D, On_Stasis)

if Y==-1:
   print "Den yparxei to onoma"
else:
   print "Eiserxomenoi =", A[Y], "Exerxomenoi=", B[Y]
```

1) Ένα ξενοδοχείο διαθέτει 50 δωμάτια και λειτουργεί 100 ημέρες τον χρόνο. Θεωρείστε ότι το κόστος διανυκτέρευσης είναι 80€ για κάθε δωμάτιο.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμούPython το οποίο να πραγματοποιεί τα παρακάτω:

Γ1. Για κάθε μέρα να διαβάζει το πλήθος των αναχωρήσεων (δωμάτια που αδειάζουν) ελέγχοντας την εγκυρότητα των δεδομένων, δηλαδή, ότι το πλήθος των αναχωρήσεων είναι μικρότερο ή ίσο από το πλήθος των κατειλημμένων δωματίων.

Γ2. Για κάθε μέρα να διαβάζει το πλήθος των αφίξεων (δωμάτια που γεμίζουν) ελέγχοντας την εγκυρότητα των δεδομένων, δηλαδή, ότι το πλήθος των αφίξεων δεν μπορεί να είναι μεγαλύτερο από το πλήθος των κενών δωματίων.

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος των ημερών που το ξενοδοχείο είχε πληρότητα 100%, δηλαδή και τα πενήντα (50) δωμάτια ήταν κατειλημμένα.

Γ4. Να υπολογίζει και να εμφανίζει τον μέσο όρο του πλήθους των δωματίων (μέση πληρότητα) που ήταν κατειλημμένα στο διάστημα λειτουργίας των εκατό (100) ημερών.

Γ5. Να υπολογίζει και να εμφανίζει τα συνολικά έσοδα του ξενοδοχείου από τις διανυκτερεύσεις για το διάστημα λειτουργίας των εκατό (100) ημερών.

2) Μία τράπεζα, για τις ανάγκες εξυπηρέτησης των πελατών της, διατηρεί δύο παράλληλες ταξινομημένες λίστες κατά αύξουσα σειρά με βάση τον αριθμό λογαριασμού. Μία λίστα με όνομα LOG[ ], η οποία περιλαμβάνει τους αριθμούς λογαριασμών των πελατών της, και μία λίστα με όνομα ΚΑΤΑΤΗ[ ], η οποία περιλαμβάνει το υπόλοιπο του λογαριασμού κάθε πελάτη. Θεωρείστε ότι όλοι οι αριθμοί λογαριασμών που υπάρχουν στη λίστα LOG[ ] είναι μοναδικοί θετικοί ακέραιοι.

Η τράπεζα επιθυμεί να γνωρίζει το ύψος των καταθέσεων και των αναλήψεων που πραγματοποιήθηκαν κατά τη διάρκεια λειτουργίας του ταμείου.

Να γράψετε τμήμα προγράμματος και τις απαραίτητες συναρτήσεις σε γλώσσα προγραμματισμούPython που να πραγματοποιεί τα παρακάτω:

Δ1. Να διαβάζει για κάθε αιτούμενη συναλλαγή τον αριθμό λογαριασμού, το είδος της συναλλαγής(1-κατάθεση, 2-ανάληψη) και το ποσό συναλλαγής. Η επαναληπτική διαδικασία τερματίζεται (κλείσιμο του ταμείου), όταν δοθεί ως αριθμός λογαριασμού ο αριθμός μηδέν (0).

Δ2. Να αναζητά τον αριθμό λογαριασμού στη λίστα LOG[ ] και να εμφανίζει το τρέχον υπόλοιπό του από τη λίστα ΚΑΤΑΤΗ[ ].

Δ3. Να υλοποιήσετε συνάρτηση money(), η οποία θα καλείται σε περίπτωση που η αιτούμενη συναλλαγή είναι ανάληψη, ώστε να ελέγξει αν επαρκεί το υπόλοιπο του λογαριασμού για να πραγματοποιηθεί η συναλλαγή. Αν το υπόλοιπο επαρκεί, να γίνεται η ανάληψη, σε διαφορετική

περίπτωση να ενημερώνεται ο πελάτης με κατάλληλο μήνυμα. Αν η αιτούμενη συναλλαγή είναι κατάθεση, να πραγματοποιείται η συναλλαγή.

Δ4. Να υπολογίζει και να εμφανίζει το άθροισμα των καταθέσεων και των αναλήψεων, μετά το κλείσιμο του ταμείου.

3) Σε έναν αθλητικό-μαθητικό αγώνα στίβου, στο αγώνισμα του μήκους συμμετέχουν 120 μαθητές από όλα τα σχολεία μιας περιφέρειας. Στον τελικό περνούν όσοι μαθητές σημειώνουν επίδοση μεγαλύτερη ή ίση των 4,5 m. Κάθε αθλητής έχει τρεις προσπάθειες για να προκριθεί. Αν σε κάποια από αυτές σημειώσει επίδοση ίση ή μεγαλύτερη από το όριο πρόκρισης, σταματάει τις προσπάθειές του.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω:

Γ1. Να διαβάζει το επώνυμο του μαθητή και να το καταχωρίζει στη λίστα ΟΝΟΜΑ των συμμετεχόντων μαθητών.

Γ2. Για κάθε μαθητή να διαβάζει την επίδοσή του ανά προσπάθεια και, αν αυτή είναι ίση ή μεγαλύτερη από το όριο πρόκρισης, να καταχωρίζει την επίδοση στη λίστα EPID και το όνομά του στη λίστα PROK των προκριθέντων μαθητών.

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος των μαθητών που προκρίθηκαν αλλά και το πλήθος των μαθητών που δεν τα κατάφεραν.

Γ4. Να εμφανίζει το επώνυμο του μαθητή με την καλύτερη επίδοση και την επίδοσή του. Αν υπάρχουν μαθητές με την ίδια επίδοση, να εμφανίζει όλα τα επώνυμα.

Σημείωση:

Δεν υπάρχουν μαθητές με το ίδιο επώνυμο.

ΑΣΚΗΣΗ 1

```
K=0    #Gemata domatia
S=0.0
c=0
for  x  in range(100):
   if  K!=0:
      AN=input('Dose arithmo anaxoriseon')
      while AN > K :
         AN=input('Dose SOSTO arithmo anaxoriseon')
      K=K-AN
   AF=input('Dose arithmo afikseon')
   while AF > 50 - K :
      AF=input('Dose SOSTO arithmo afikseon')
   K=K+AF
   if K==50:
      c=c+1
   S=S+K

print "plithos hmeron me plirotita 100%=", c

print "Mesi plirotita =", S/100

print "Synolika esoda = ", S*80
```

ΑΣΚΗΣΗ 2

```
def money(Y, p):
   if p>Y:
      print "To diathesim ypoloipo den eparkei"
      return 1
   else:
      return 0

LOG=[]
KATATH=[]
SA=0
SK=0

AL=input('Dose arithmo logariasmou')
while AL!=0:
   for x  in range(len(LOG)):
      if AL==LOG[x]:
         print KATATH[x]
         ES=input('Dose eidos synalagis')
         p=input('Dose poso synalagis')
         if ES==1:
            KATATH[x]=KATATH[x] + p
            SK=SK+p
         if ES==2:
            V=money(KATATH[x], p)
            if V==0:
               KATATH[x]=KATATH[x] - p
               SA=SA+p

   AL=input('Dose arithmo logariasmou')

print "Synolikes katatheseis : ", SK
print "Synolikes analipseis : ", SA
```

ΑΣΚΗΣΗ 3

```
ONOMA=[]
EPID=[]
PROK=[]

for  x  in range(120):
   p=1
   on=raw_input('Dose onoma athliti')
   ONOMA.append(on)
   ep=input('Dose epidosi')
   while ep<4.5 and p<=3:
      p=p+1
      ep=input('Dose kaliteri epidosi')
   if   ep>=4.5:
      EPID.apend(ep)
      PROK.append(on)
print "prokrithikan : ", len(PROK)
print "den prokrithikan : ", 120-len(PROK)

max1=EPID[0]
for x in EPID:
   if x > max1:
      max1=x

for  x  in range(len(EPID)):
   if   EPID[x]==max1:
      print PROK[x],   EPID[x]
```

1) Μια εταιρία κινητής τηλεφωνίας χρεώνει τους πελάτες της, ως προς τον όγκο κατανάλωσης δεδομένων, κλιμακωτά, σύμφωνα με τον παρακάτω πίνακα:

Όγκος Κατανάλωσης σε MB	**Χρέωση ανά MB**
Τα πρώτα 50	0,03ευρώ
Από 51 μέχρι και 200	0,02 ευρώ
Περισσότερα από 200	0,01 ευρώ

Ένας καταναλωτής αγοράζει από την εταιρία 2000 MB τα οποία **λήγουν σε 20 ημέρες**.

Κάθε μέρα καταναλώνει κάποια από τα MB.

Να δίνετε από το πληκτρολόγιο για κάθε μέρα τα MB που κατανάλωσε μέχρι να ξοδευτούν όλα.

Α) Να υπολογιστεί και να τυπωθεί το συνολικό ποσό που θα πληρώσει ο καταναλωτής

Β) Σε περίπτωση που λήξουν τα MB πόσα MB **δεν χρησιμοποίησε** και πόσο κοστίζουν

Γ) Ποια ημέρα θα πληρώσει το μεγαλύτερο ποσό χρημάτων και πόσο είναι αυτό

Σημείωση: Να χρησιμοποιηθεί **συνάρτηση** για τον υπολογισμό των χρημάτων στην κλιμακωτή χρέωση

ΑΣΚΗΣΗ 1

```
def   klimakoti(k):
   if k>=1 and k<=50:
      xr=k*0.03
   if k>=51 and k<=200:
      xr=50*0.03 + (k-50)* 0.02
   if k>200:
      xr=50*0.03 + 150 * 0.02 + (k-200)*0.01
   return xr

H=[]
XR=[]
k=input('Dose katanalosi')
S=0
Sxr=0
h=1
while S+k<=2000 and h<=20:
   S=S+k
   xr=klimakoti(k)
   H.append(k)
   XR.append(xr)
   k=input('Dose nea katanalosi')
   h=h+1

S=0
for x in XR:
   S=S+x

print "Synoliko poso pliromis :", S

if S<2000:
   print "Ypoloipa MB :", 2000-S, "kai kostizoun", klimakoti(2000-S)

max1=XR[0]
h=0
for x in range(1,len(XR)):
   if XR[x]>max1:
      max1=XR[x]
      h=x

print "Megalitero poso tin hmera :", h+1, "kai einai ", max1
```

1) Έστω ότι ο υπολογισμός φόρου φυσικών προσώπων γίνεται με βάση τον πίνακα:

ΕΙΣΟΔΗΜΑ	**ΣΥΝΤΕΛΕΣΤΗΣ ΦΟΡΟΥ**
Μέχρι και 5.000	0%
Από 5.001 μέχρι 8.000	10%
Από 8.001 μέχρι 12.000	15%
Άνω των 12.001	20%

{δηλαδή αν κάποιος έχει εισόδημα 10.000 € για τις πρώτες 5.000 δεν πληρώνει τίποτα, για το τμήμα από 5.000 μέχρι 8.000 θα πληρώσει 10% και για τις υπόλοιπες 2.000 15%}.

Επίσης αν ο φορολογούμενος έχει παιδιά αφαιρούνται από το φόρο του 60 € για κάθε παιδί μέχρι και το 3^{ο} , και 200 € συνολικά αν έχει πάνω από 3 (για όλα μαζί).

Να φτιάξετε πρόγραμμα που διαβάζει το ΑΦΜ, το εισόδημα και τον αρ. παιδιών ενός φορολογουμένου, να υπολογίζει και να εμφανίζει το εισόδημά του, τον αρ. παιδιών και το φόρο που θα πληρώσει (ο φόρος δεν μπορεί να είναι αρνητικός). Η διαδικασία να επαναλαμβάνεται μέχρι να δοθεί ΑΦΜ 0.

ΑΣΚΗΣΗ 1

```
AFM=input('Dose AFM')

while AFM !=0:
   E=input('Dose eisodima')
   P=input('arithmos paidion')

   if E>=1 and E<=5000:
      F=0
   if E>=5001 and E<=8000:
      F= (E-5000) * 10 / 100
   if E>=8001 and E<=12000:
      F= 3000 * 10 / 100 + (E- 8000) * 15/100
   if E> 12000:
      F= 3000 * 10 / 100 + 4000 * 15/100 + (E- 12000)* 20/100

   if P>=1 and P<=3:
      F=F-60*P

   if P>3:
      F=F-200

   Xr=E-F
   print "teliko eisodima : ", Xr

   AFM=input('Dose AFM')
```

1) Ένα παρκινγκ διαθέτει 120 θέσεις και χρεώνει κλιμακωτά τη στάθμευση σε αυτές σύμφωνα με τον παρακάτω πίνακα:

Ώρες στάθμευσης	**Κόστος(€)**
Λιγότερες από 3	2.5
Από 3 έως λιγότερες από 6	1.5
Από 6 ώρες έως λιγότερες από 9	1
Για τις επιπλέον ώρες το κόστος είναι 10 € για όλες τις ώρες	

Για παράδειγμα, αν ένα αυτοκίνητο έμεινε 4 ώρες θα πληρώσει 8 €, ενώ αν διέμεινε 7 ώρες θα πληρώσει 11.5 €.

Να κατασκευάσετε αλγόριθμο, ο οποίος:

α) για κάθε αυτοκίνητο που στάθμευσε στο παρκινγκ να διαβάζει τον αριθμό κυκλοφορίας του και τη διάρκεια στάθμευσης σε ώρες, την οποία να δέχεται μόνο εφ’ όσον είναι μεγαλύτερη από το 0. Θεωρούμε ότι το παρκινγκ γέμισε και κάθε θέση καταλήφθηκε μόνο μια φορά από κάποιο αυτοκίνητο.

β) να υπολογίζει το ποσό που πρέπει να πληρώσει ο κάτοχός του.

γ) να εμφανίζει τον αριθμό κυκλοφορίας και το ποσό που αναλογεί.

δ) να εμφανίζει τις συνολικές εισπράξεις του παρκινγκ.

ε) να εμφανίζει το ποσοστό των αυτοκινήτων που στάθμευσαν περισσότερες από 3 ώρες στο παρκινγκ.

στ) Αν κάθε αυτοκίνητο στάθμευε στο παρκινγκ για 3 ώρες, να εμφανίζεται μήνυμα σχετικά με το αν τα έσοδά του θα ήταν περισσότερα, λιγότερα ή ίσα με τις πραγματικές εισπράξεις που πραγματοποιήθηκαν.

ΑΣΚΗΣΗ 1

```
S=0
c=0
for  x  in range(120):
   P=raw_input('Dose pinakida')
   Or=input('Dose ores')
   while Or<=0:
      Or=input('Dose SOSTES ores')

   if Or>=1 and Or<=2:
      Xr=Or * 2.5
   if Or>=3 and Or<=5:
      Xr= 2 * 2.5 + (Or-2)*1.5
   if Or>=6 and Or<=8:
      Xr= 2 * 2.5 + 3 * 1.5 + (Or-5)*1

   if  Or>=9 :
      Xr= 10

   print  P, Xr

   S=S+Xr

   if Or>3:
      c=c+1

print "Synolikes eisprakseis :", S

print "Pososto pano apo 3 ores : ", 100*c/120.0

Xr2= (2 * 2.5 + 1.5) *120

if Xr2 > S:
   print "Perisoteres eisprakseis gia 3 ores"

else:
   print "ligoteres eisprakseis gia 3 ores"
```

1) Στο κέντρο συγκέντρωσης προσφύγων Μόρια, στην Λέσβο, για την προμήθεια νερού εστάλη ένα μεγάλο βυτιοφόρο που διαθέτει δεξαμενή χωρητικότητας 50.000 λίτρων. Να γράψετε πρόγραμμα Python το οποίο:

Γ1. να διαβάζει την αρχική ποσότητα του νερού(σε λίτρα) που υπάρχει στο βυτιοφόρο. Να γίνει έλεγχος εγκυρότητας.

Για κάθε πρόσφυγα που προσέρχεται να πάρει νερό:

Γ2. να διαβάζει την εθνικότητα του "ΣΥΡΙΟΣ" ή "ΑΛΛΟΣ" και το φύλο του "Α"=ΑΝΔΡΑΣ, "Γ"=ΓΥΝΑΙΚΑ και "Π"=ΠΑΙΔΙ. Να μην γίνει έλεγχος εγκυρότητας.

Γ3. Να διαβάζει την χωρητικότητα του μπιτονιού που έχει ο πρόσφυγας σε λίτρα και να του το γεμίζει αν υπάρχει επάρκεια νερού στο βυτίο, διαφορετικά να μην εξυπηρετείται.

Γ4. Η επαναληπτική διαδικασία να τερματίζεται, όταν αδειάσει το βυτίο ή όταν δεν εξυπηρετηθούν τρία διαδοχικά άτομα.

Γ5. Στο τέλος το πρόγραμμα, αφού έχει υπολογίσει πριν, να εμφανίζει:

- τη μέση ποσότητα νερού ανά φύλλο που εξυπηρετήθηκε.
- τη συνολική ποσότητα νερού που διατέθηκε σε ΣΥΡΙΑ παιδιά.
- Ποιας εθνικότητας και φύλλο ήταν ο πρόσφυγας που πήρε το περισσότερο νερό.
- το πλήθος των Σύριων και το πλήθος των άλλων που εξυπερετήθηκαν.

Σημείωση:
Θεωρήστε ότι στο βυτίο προσέρχεται ένας τουλάχιστον πρόσφυγας για τον οποίο η ποσότητα νερού επαρκεί. Επίσης θεωρείστε την χωρητικότητα του κάθε μπιτονιού θετική.

ΑΣΚΗΣΗ 1

```
dexyp = 0          # δεν εξυπηρετήθηκαν
pna = 0.0          # ποσότητα νερού σε άνδρες που εξυπηρετήθηκαν
png = 0.0          # ποσότητα νερού σε γυναίκες που εξυπηρετήθηκαν
pnp = 0.0          # ποσότητα νερού σε παιδιά που εξυπηρετήθηκαν
pla = 0           # πλήθος ανδρών που εξυπηρετήθηκαν
plg = 0           # πλήθος γυναικών που εξυπηρετήθηκαν
plp = 0           # πλήθος παιδιών που εξυπηρετήθηκαν
plsyr = 0          # πλήθος ΣΥΡΙΩΝ που εξυπηρετήθηκαν
plal = 0          # πλήθος ΑΛΛΩΝ που εξυπηρετήθηκαν
maxn = -1          # μέγιστη ποσότητα νερού που δόθηκε σε πρόσφυγα
snsp = 0.0         # συνολική ποσότητα νερού που δόθηκε σε ΣΥΡΙΑ ΠΑΙΔΙΑ

apnv = input('Δώσε αρχική ποσότητα νερού σε λίτρα > 0 και <=50000 : ')
while apnv <= 0 or apnv > 50000:
   apnv = input('Δώσε σωστή ποσότητα > 0 και <=50000 :')

pnv = apnv         # ποσότητα νερού στο βυτίο διαθέσιμη

while pnv > 0 and dexyp < 3:
   ethn = raw_input('Δώσε εθνικότητα ΣΥΡΙΟΣ ή ΑΛΛΟΣ μόνο: ')
   fylo = raw_input('Δώσε φύλο Α=ΑΝΔΡΑΣ, Γ=ΓΥΝΑΙΚΑ ή Π=ΠΑΙΔΙ μόνο: ')
   xbit = input('Δώσε χωρητικότητα μπιτονιού σε λίτρα: ')

   if pnv >= xbit:

      pnv = pnv – xbit
      dexyp = 0

   if ethn == “ΣΥΡΙΟΣ”:
      plsyr = plsyr + 1
   else:
      plal = plal + 1

   if xbit > maxn:
      maxn = xbit
      maxe = ethn
      maxf = fylo

   if fylo == “Α”:
      pna = pna + xbit
      pla = pla + 1
   elif fylo == “Γ”:
      png = png + xbit
      plg = plg + 1
   elif fylo == “Π”:
      pnp = pnp + xbit
      plp = plp + 1
      if ethn == “ΣΥΡΙΟΣ”:
```

```
        snsp = snsp + xbit
   else:
      dexyp = dexyp + 1

print “μέση ποσότητα νερού ανδρών”, pna/pla
print “μέση ποσότητα νερού γυναικών”, png/plg
print “μέση ποσότητα νερού παιδιών”, pnp/plp

print “συνολική ποσότητα νερού σε ΣΥΡΙΑ παιδιά”, snsp

print “πρόσφυγας με περισσότερο νερό ήταν εθνικότητας”, maxe, “και φύλλο”, maxf

print “Συνολικά εξυπηρετήθηκαν ΣΥΡΙΟΙ=”, plsyr, “και ΑΛΛΟΙ=”, plal
```

1) Στο Πανευρωπαϊκό πρωτάθλημα στίβου το 2016 στον τελικό το αθλήματος Σφύρας αγωνίστηκαν 16 αθλητές. Κάθε αθλητής είχε στην διάθεσή του έξι προσπάθειες. Από τις 6 προσπάθειες στην τελική του επίδοση μετράει αυτή στην οποία πέταξε την σφύρα στα περισσότερα μέτρα.

Να γράψετε πρόγραμμα python το οποίο:

Δ1. Θα καταγράφει σε λίστα on τα ονόματα των 16 αθλητών, σε λίστα χο την χώρα προέλευσης του αθλητή και σε λίστα ep την μεγαλύτερη ρίψη που έκανε ο κάθε αθλητής.

Δ2. Θα διαβάζει το όνομα ενός αθλητή και την χώρα του. Με χρήση κατάλληλου υποπρογράμματος(Δ4), να ελέγχει αν αυτός υπάρχει στην λίστα. Αν υπάρχει να εκτυπώνει την επίδοσή του. Αν δεν υπάρχει να τυπώνει κατάλληλο μήνυμα.

Δ3. Να τυπώνει τους αθλητές (όνομα, χώρα, επίδοση), με φθίνουσα σειρά ως προς την επίδοση τους.

Δ4. Το υποπρόγραμμα θα δέχεται σαν παραμέτρους δύο λίστες list1, list2 και δύο κλειδιά key1, key2. Αν το key1 υπάρχει στην list1 σε κάποια θέση και στην αντίστοιχη(ίδια) θέση της list2 υπάρχει το key2 να επιστρέφει την θέση που το βρήκε αλλιώς να επιστρέφει την τιμή -1. Θεωρείστε ότι υπάρχει μία μόνο τέτοια σύμπτωση.

ΑΣΚΗΣΗ 1

```
def search(list1,list2,key1,key2):
   thesi = -1
   for i in range(0,n,1):
      if (key1 == list1[i]) and (key2 == list2[i]):
         thesi = i
   return thesi

n=16
on = []
xo = []
ep = []


for i in range(0,n):
   on.append(raw_input('Δώσε όνομα αθλητή: '))
   xo.append(raw_input('Δώσε χώρα αθλητή: '))
   ep.append(input('Δώσε μέγιστη βολή σε μέτρα: '))

print 'on=',on
print 'xo=',xo
print 'ep=',ep

onoma = raw_input('Δώσε το όνομα ενός αθλητή: ')
xora = raw_input('Δώσε την χώρα του: ')
t = search(on,xo,onoma,xora)
if t == -1:
   print 'ο αθλητής δεν βρέθηκε'
else:
   print 'αθλητής',on[t],'χώρα',xo[t],'επίδοση',ep[t]
for i in range(1 , n, 1):
   for j in range(n-1, i-1 , -1):
      if ep[j] > ep[j-1] :
         ep[j], ep[j-1] = ep[j-1], ep[j]
         on[j], on[j-1] = on[j-1], on[j]
         xo[j], xo[j-1] = xo[j-1], xo[j]

print 'Κατάταξη σε φθίνουσα σειρά ως προς την επίδοση'

for i in range (0,n,1):
   print on[i], xo[i], ep[i]
```

1) Μία εταιρεία διαθέτει 30 υπαλλήλους και ανάλογα με το βασικό μισθό που παίρνουν τους γίνονται οι παρακάτω κρατήσεις.

ΜΙΣΘΟΣ σε ευρώ	**ΠΟΣΟΣΤΟ ΚΡΑΤΗΣΕΩΝ**
1-1000	10%
Πάνω από 1000 – 2000	15%
Πάνω από 2000	20%

Οι κρατήσεις **δεν γίνονται κλιμακωτά**. Δηλαδή για παράδειγμα αν κάποιος έχει βασικό μισθό 3000 €, θα έχει κρατήσεις 3000*20%=600 €

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Να διαβάζει το ονοματεπώνυμο και το βασικό μισθό του κάθε υπαλλήλου και να υπολογίζει και να εμφανίζει το ποσό κρατήσεων και τον τελικό μισθό που πρόκειται να πάρει. Να γίνει έλεγχος ορθότητας στο βασικό μισθό, ότι δηλαδή ο βασικός μισθός που καταχωρίζεται είναι από 1 έως και 10000.

Γ2. Να υπολογίζει και να εμφανίζει το συνολικό ποσό κρατήσεων και το μέσο όρο κρατήσεων για όλους τους υπαλλήλους

Γ3. Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του υπαλλήλου με το μικρότερο ποσό κρατήσεων και το ποσό αυτό.

Γ4. Να υπολογίζει και να εμφανίζει το πλήθος των υπαλλήλων που ανήκει σε κάθε μία από τις παραπάνω περιπτώσεις.

2) Μία εταιρία διαθέτει 3 είδη ιστιοπλοϊκών το Α το Β και το Γ. Το κόστος ενοικίασης του καθενός για κάθε ημέρα φαίνεται στον παρακάτω πίνακα:

ΤΥΠΟΣ ΙΣΤΙΟΠΛΟΪΚΟΥ	**ΚΟΣΤΟΣ ΑΝΑ ΗΜΕΡΑ**
Α	200 €
Β	300 €
Γ	400 €

Να γράψετε πρόγραμμα σε Python το οποίο:

Δ1. Να διαβάζει τον τύπο του ιστιοπλοϊκού και τις ημέρες που ενοικιάστηκε και να υπολογίζει και να εμφανίζει το κόστος ενοικίασης. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως τύπος ιστιοπλοϊκού η λέξη «TELOS». Στη συνέχεια να τοποθετεί τον τύπο σε μία λίστα με το όνομα TYPOS και τις ημέρες σε μία λίστα με το όνομα HMERES. (Δεν απαιτείται έλεγχος εγκυρότητας εισαγωγής των δεδομένων.)

Δ2. Να υπολογίζει και να εμφανίζει το σύνολο των ημερών που νοικιάστηκε κάθε τύπος. Για παράδειγμα:

Α 23

Β 34

Γ 45

(Οι παραπάνω τιμές, όπως και ο τρόπος εμφάνισης-στοίχισης δίνονται ενδεικτικά.)

Δ3. Να εγγράφει σε ένα αρχείο synola.txt τα στοιχεία που εμφανίζονται στο υποερώτημα Δ2, όπως ακριβώς φαίνονται στο παράδειγμα, δηλαδή σε κάθε μία γραμμή να γραφεί ο τύπος και οι συνολικές ημέρες ενοικίασης.

Δ4. Να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της ευθείας ανταλλαγής (φυσαλίδα-bubble sort) τις δύο λίστες σε αύξουσα σειρά ως προς τους τύπους ώστε να τοποθετηθούν όλα τα Α μαζί, όλα τα Β μαζί και όλα τα Γ μαζί .

Δ5. Να εγγράφει στο αρχείο synola.txt , χωρίς να σβηστούν τα προηγούμενα περιεχόμενα, τα περιεχόμενα των δύο λιστών, έχοντας σε κάθε γραμμή τον τύπο και τις ημέρες που ενοικιάστηκε όπως φαίνεται στο παρακάτω παράδειγμα:

π.χ.
Α 6
Α 3
Α 4
Β 4
Β 7
Γ 2
κλπ

ΑΣΚΗΣΗ 1

```
SUMK=0.0
MINK=10000
MINON=""

m1=m2=m3=0

for i in range(30):
   on=raw_input("Δώσε το ονοματεπώνυμο")
   M=input("Δώσε το βασικό μισθό")
   while not(M>=1 and M<=10000):
      M=input("Δώσε το βασικό μισθό")

   if M>=1 and M<=1000:
      pk=M*10/100
      m1=m1+1
   elif M>1000 and M<=2000:
      pk=M*15/100
      m2=m2+1
   elif M>2000:
      pk=M*20/100
      m3=m3+1

   tm=M-pk

   print "Ο υπάλληλος ",on," έχει ποσό κρατήσεων ",pk,"και τελικο μισθό ",tm
   SUMK=SUMK+pk

   if pk<MINK:
      MINK=pk
      MINON=on

MOK=SUMK/30
print "Σύνολο κρατήσεων",SUMK
print "Μέσος όρος κρατήσεων",MOK
print "Ο Υπάλληλος ",MINON,"έχει τις ελάχιστε κρατήσεις που είναι ",MINK
print "Στην κατηγορία με βασικό μισθό από 1-1000 έχουμε",m1,'υπαλλήλους'
print "Στην κατηγορία με βασικό μισθό πάνω από 1000-2000 έχουμε",m2,'υπαλλήλους'
print "Στην κατηγορία με βασικό μισθό πάνω από 2000 έχουμε",m3,'υπαλλήλους'
```

ΑΣΚΗΣΗ 2

```
def bubblesort(A,B):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
TYPOS=[]
HMERES=[]
sumA=sumB=sumG=0

typos=raw_input("Δώσε τον τύπο του ιστιοπλοϊκού")
while typos!="TELOS":
   h=int(input("Δώσε τις ημέρες που ενοικιάστηκε"))
   if typos=="A":
      ke=h*200
      sumA=sumA+h
   elif typos=="B":
      ke=h*300
      sumB=sumB+h
   elif typos=="Γ":
      ke=h*400
      sumG=sumG+h
   print "Το κόστος ενοικίασης είναι",ke
   TYPOS.append(typos)
   HMERES.append(h)
   typos=raw_input("Δώσε τον τύπο του ιστιοπλοϊκού")


print "Το σύνολο ημερών που ενοικιάστηκε ο κάθε τύπος είναι "
print "A",sumA
print "B",sumB
print "Γ",sumG

f=open("synola.txt","w")
f.write("A" + " " + str(sumA) + "\n")
f.write("B" + " " + str(sumB) + "\n")
f.write("Γ" + " " + str(sumG) + "\n")
f.close()

bubblesort(TYPOS,HMERES)

f=open("synola.txt","a")
for i in range(len(TYPOS)):
   f.write(TYPOS[i] + " " + str(HMERES[i]) + "\n")
f.close()
```

1) Μία εταιρία διαθέτει 3 είδη ιστιοπλοϊκών το Α το Β και το Γ. Το κόστος ενοικίασης του καθενός για κάθε ημέρα φαίνεται στον παρακάτω πίνακα:

ΤΥΠΟΣ ΙΣΤΙΟΠΛΟΪΚΟΥ	**ΚΟΣΤΟΣ ΑΝΑ ΗΜΕΡΑ**
Α	200 €
Β	300 €
Γ	400 €

Να γράψετε πρόγραμμα σε Python το οποίο:

Δ1. Να διαβάζει τον τύπο του ιστιοπλοϊκού και τις ημέρες που ενοικιάστηκε και να υπολογίζει και να εμφανίζει το κόστος ενοικίασης. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως τύπος ιστιοπλοϊκού η λέξη «TELOS». Στη συνέχεια να τοποθετεί τον τύπο σε μία λίστα με το όνομα TYPOS και τις ημέρες σε μία λίστα με το όνομα HMERES. (Δεν απαιτείται έλεγχος εγκυρότητας εισαγωγής των δεδομένων.)

Δ2. Να υπολογίζει και να εμφανίζει το σύνολο των ημερών που νοικιάστηκε κάθε τύπος. Για παράδειγμα:

Α 23

Β 34

Γ 45

(Οι παραπάνω τιμές, όπως και ο τρόπος εμφάνισης-στοίχισης δίνονται ενδεικτικά.)

Δ3. Να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της ευθείας ανταλλαγής (φυσαλίδα-bubble sort) τις δύο λίστες σε αύξουσα σειρά ως προς τους τύπους ώστε να τοποθετηθούν όλα τα Α μαζί, όλα τα Β μαζί και όλα τα Γ μαζί .

Δ4. Να δημιουργηθεί μία λίστα με όνομα ALL η οποία θα περιέχει τρείς υπολίστες : α) την υπολίστα που θα περιέχει τις ημέρες του τύπου Α , β) την υπολίστα που θα περιέχει τις ημέρες του τύπου Β και γ) την υπολίστα που θα περιέχει τις ημέρες του τύπου Γ.

Δ5. Να δίνετε από το πληκτρολόγιο τον τύπο ενός ιστιοπλοϊκού και χρησιμοποιώντας μόνο την λίστα ALL να τυπώνονται οι ημέρες του συγκεκριμένου τύπου.

ΑΣΚΗΣΗ 1

```
def bubblesort(A,B):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
TYPOS=[ ] ;    HMERES=[ ]   ;  sumA=sumB=sumG=0

typos=raw_input("Δώσε τον τύπο του ιστιοπλοϊκού")
while typos!="TELOS":
   h=int(input("Δώσε τις ημέρες που ενοικιάστηκε"))
   if typos=="A":
      ke=h*200
      sumA=sumA+h
   elif typos=="B":
      ke=h*300
      sumB=sumB+h
   elif typos=="Γ":
      ke=h*400
      sumG=sumG+h
   print "Το κόστος ενοικίασης είναι",ke
   TYPOS.append(typos)
   HMERES.append(h)
   typos=raw_input("Δώσε τον τύπο του ιστιοπλοϊκού")

print "Το σύνολο ημερών που ενοικιάστηκε ο κάθε τύπος είναι "
print "A",sumA
print "B",sumB
print "Γ",sumG

ALL=[] ;  A=[]  ; B=[]  ;C=[]
for x in range(len(TYPOS)):
   if TYPOS[x]=="A":
      A.append( HMERES[x])
   if TYPOS[x]=="B":
      B.append(HMERES[x])

   if TYPOS[x]=="Γ":
      C.append(HMERES[x])

ALL=[A,B,C]
T=raw_input('Dose typo')
if  T=="A":
   for x in range(len(ALL[0])):
      print ALL [0] [x]
if  T=="B":
   for x in range(len(ALL[1])):
      print ALL [1] [x]
if  T=="Γ":
   for x in range(len(ALL[2])):
      print ALL [2] [x]
```

1) Μία εταιρεία διαθέτει δύο διαμερίσματα για ενοικίαση: το διαμέρισμα με αριθμό 1 και αυτό με τον αριθμό 2. Ανάλογα με το πόσες βραδιές θα νοικιάσει κάποιος ένα διαμέρισμα, εφαρμόζεται Κλιμακωτή Χρέωση σύμφωνα με τον παρακάτω πίνακα:

Ημέρες	**Διαμέρισμα με αριθμό 1 (κόστος ανά ημέρα)**	**Διαμέρισμα με αριθμό 2 (κόστος ανά ημέρα)**
1-5	150 €	180 €
6-15	120 €	140 €
16 και πάνω	100 €	110 €

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Να διαβάζει τον αριθμό του διαμερίσματος που θέλει να νοικιάσει και τις ημέρες που θέλει να μείνει. Η διαδικασία αυτή να επαναλαμβάνεται για 20 πελάτες. (Δεν απαιτείται έλεγχος εγκυρότητας εισαγωγής των δεδομένων.)

Γ2. Να υπολογίζει το κόστος διαμονής για κάθε πελάτη και να το εμφανίζει με κατάλληλο μήνυμα.

Γ3. Να υπολογίζει και να εμφανίζει το σύνολο των ημερών που έμεινε νοικιασμένο το κάθε ένα διαμέρισμα χωριστά.

Γ4. Να υπολογίζει και να εμφανίζει το σύνολο των ημερών που νοικιάστηκαν και τα δύο διαμερίσματα.

2) Σε ένα θέατρο παίζεται αυτή την περίοδο μία συγκεκριμένη θεατρική παράσταση την οποία μπορεί να παρακολουθήσουν μαθητές, από διάφορα σχολεία, με εισιτήριο 5 ευρώ ο καθένας. Η χωρητικότητα του θεάτρου είναι 300 άτομα.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. Να διαβάζει την ονομασία του Σχολείου και το πλήθος των μαθητών που πρόκειται να δουν την παράσταση. Τα στοιχεία αυτά καταχωρίζονται στις λίστες NAME και PLITHOS αντίστοιχα. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως ονομασία Σχολείου η λέξη «TELOS» . Να γίνει έλεγχος ορθότητας ότι δηλαδή το πλήθος των μαθητών που καταχωρίζεται είναι από 1 έως και 300.

Δ2. Να υπολογίζει και να εμφανίζει το κόστος για κάθε ένα σχολείο.

Δ3. Να υπολογίζει και να εμφανίζει το συνολικό κόστος για όλα τα σχολεία .

Δ4. Να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της ευθείας ανταλλαγής (φυσαλίδα-bubble sort) τις δύο λίστες σε αύξουσα σειρά ως προς το πλήθος των μαθητών .

Δ5. Να υπολογίζει και να εμφανίζει τις ονομασίες των 3 σχολείων με τους λιγότερους μαθητές, θεωρώντας ότι δεν υπάρχουν σχολεία με ίσο αριθμό μαθητών και ότι μας δόθηκαν τουλάχιστον 3 σχολεία.

ΑΣΚΗΣΗ 1

```
sum1=0
sum2=0
sumT=0
for i in range(20):
   ar=int(input("Δώσε τον αριθμό του Διαμερίσματος"))
   hm=int(input("Δώσε τον αριθμό των ημερών που θέλεις να μείνεις"))
   if ar==1:
      sum1=sum1+hm
      if hm>=1 and hm<=5:
         kostos=hm*150
      if hm>=6 and hm<=15:
         kostos=5 *150 +(hm-5)*120
      if hm>16:
         kostos=5*150+10*120+(hm-15)*100

   if ar==2:
      sum2=sum2+hm #Γ3
      if hm>=1 and hm<=5:
         kostos=hm*180
      if hm>=6 and hm<=15:
         kostos=5 *180 +(hm-5)*140
      if hm>16:
         kostos=5*180+10*140+(hm-15)*110

   print "Το κόστος διαμονής στο διαμέρισμα ",ar,"για ",hm,"ημέρες ειναι ",kostos,"€"
print "Το διαμέρισμα 1 νοικιάστηκε για ",sum1,"ημέρες συνολικά"
print "Το διαμέρισμα 2 νοικιάστηκε για ",sum2,"ημέρες συνολικά"
sumT=sum1+sum2
print "Και τα δύο διαμερίσματα νοικιάστηκαν για ", sumT, "ημέρες συνολικά"
```

ΑΣΚΗΣΗ 2

```
def bubblesort(A,B):
   N=len(A)
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
NAME=[]
PLITHOS=[]
SUM=0
on=raw_input("Δώσε την ονομασία του σχολείου")
while on!="TELOS":
   p=int(input("Δώσε το πλήθος των μαθητών που θα δουν την παράσταση"))
   while p<1 or p>300:
      p=int(input("Δώσε το πλήθος των μαθητών που θα δουν την παράσταση"))

   PLITHOS.append(p)
   kostos=5*p
   print "Το κόστος για το σχολείο ",on," είναι",kostos,"€."
   SUM=SUM+kostos
   on=raw_input("Δώσε την ονομασία του σχολείου")

print "Το συνολικό κόστος είναι",SUM,"€."
bubblesort(PLITHOS,NAME)
print "Τα 3 σχολεία με τους λιγότερους μαθητές είναι:"
print "Το σχολείο ", NAME[0], "με ",PLITHOS[0]," μαθητές"
print "Το σχολείο ", NAME[1], "με ",PLITHOS[1]," μαθητές"
print "Το σχολείο ", NAME[2], "με ",PLITHOS[2]," μαθητές"
```

1) Μια πολυεθνική εταιρεία λόγω της οικονομικής κρίσης αποφάσισε να κάνει περικοπές στους μισθούς των 120 υπαλλήλων της.

Να γράψετε πρόγραμμα το οποίο:

Δ1. Θα καταχωρεί σε λίστα Μ τους μισθούς των υπαλλήλων, σε λίστα ΟΝ τα ονόματα τους και σε λίστα F το φύλο τους.

Να γίνεται κατάλληλος έλεγχος εγκυρότητας ώστε:

- Ο πρώτος χαρακτήρας κάθε ονόματος δεν πρέπει να ξεκινά από γράμμα μικρότερο από Ε΄ αλλά και ούτε από γράμμα μεγαλύτερο από 'Ζ'.
- Για το φύλο επιτρεπτές τιμές είναι η 'Α΄ και 'Γ'.

Δ2. Θα καταχωρεί σε λίστα Ρ τη μείωση των μισθών των 120 υπαλλήλων εμφανίζοντας, το ποσό που θα περικοπεί από κάθε υπάλληλο σε μήνυμα της μορφής «ο μισθός του υπαλλήλου ..... θα μειωθεί κατά ..... ευρώ».

Η μείωση των μισθών θα γίνεται κλιμακωτά σύμφωνα με τον παρακάτω πίνακα:

Μισθός (ευρώ)	**Μείωση(%)**
έως και 700	3
έως και 1000	6
έως και 1500	10
πάνω από 1500	20

Δ3. Θα δημιουργεί λίστα ΝΕ η οποία θα περιέχει τους νέους μισθούς των υπαλλήλων όπως αυτοί διαμορφώθηκαν μετά την μείωση.

Δ4. Θα εμφανίζει το ποσό που θα εξοικονομήσει η εταιρεία από την περικοπή των μισθών.

Δ5. Θα ταξινομεί τα στοιχεία των υπαλλήλων με κριτήριο το νέο μισθό τους σε φθίνουσα σειρά.

Δ6. Θα εμφανίζει τα ονόματα των 10 πιο καλοπληρωμένων ανδρών.

2) Σε ένα πάρκινγκ χωρητικότητας 250 θέσεων στάθμευσης, μπορούν να σταθμεύσουν είτε αυτοκίνητα, είτε φορτηγά, με τα αυτοκίνητα να καταλαμβάνουν μία θέση και τα φορτηγά δύο θέσεις στάθμευσής. Η χρέωση για το κάθε όχημα γίνεται ανάλογα με τις ώρες παραμονής σε αυτό και σύμφωνα με τον παρακάτω πίνακα:

ΑΥΤΟΚΙΝΗΤΑ			**ΦΟΡΤΗΓΑ**	
Ώρες παραμονής	**Χρέωση ανά ώρα**		**Ώρες παραμονής**	**Χρέωση ανά ώρα**
1	2.00		Μέχρι και 2 ώρες	2.20
2	1.80		Από 3 μέχρι και 5	2.00
Πάνω από 2	1.50		Πάνω από 5 ώρες	1.80

Να γραφεί πρόγραμμα σε Python το οποίο:

Δ1. Για κάθε όχημα:

Θα διαβάζει το είδος του (Αυτοκίνητο ή Φορτηγό) και να επιτρέπεται η είσοδος στο πάρκινγκ εφόσον υπάρχει διαθέσιμος χώρος.

Αν το όχημα εισέλθει στο πάρκινγκ τότε:

Α) Να διαβάζει τις ώρες παραμονής σε αυτό.

Β) Να υπολογίζει και να εμφανίζει το ποσό σε ευρώ που θα πληρώσει ο κάτοχος του οχήματος στον ιδιοκτήτη του πάρκινγκ.

Τα παραπάνω να επαναλαμβάνονται, μέχρι το πάρκινγκ να μη χωράει άλλο όχημα.

Δ2. Όταν το πάρκινγκ δε χωράει άλλο όχημα τότε να υπολογίζει και να εμφανίζει:

Α) Τον αριθμό των αυτοκινήτων που υπάρχουν σε αυτό.

Β) Τον αριθμό των φορτηγών που υπάρχουν σε αυτό.

Γ) Το σύνολο των χρημάτων που θα πληρώσουν οι κάτοχοι όλων των οχημάτων στον ιδιοκτήτη του πάρκινγκ.

Παρατήρηση: Θεωρήστε ότι μέχρι να γεμίσει το πάρκινγκ δεν εξέρχεται κανένα όχημα από αυτό.

ΑΣΚΗΣΗ 1

```
M=[] ;  ON=[] ; F=[]  ; P=[] ; NE=[]
S=0
S1=0
for  x  in range(120):
   m=input('Dose mistho')
   on=str(raw_input('Dose onoma'))
   while on[0]<"E" or on[0]> "Z":
      on=str(raw_input('Dose SOSTO onoma'))
   f=str(raw_input('Dose fylo'))
   while f not in ["A", "Γ"] :
      f=str(raw_input('Dose SOSTO fylo'))

   M.append(m)
   ON.append(on)
   F.append(f)

   if m>=1 and m<=700 :
      p=m*3/100
   if m>=701 and m<=1000 :
      p=700*3/100 + (m-700)*6/100
   if m>=1001 and m<=1500 :
      p=700*3/100 + 300*6/100  + (m-1000)*10/100
   if m>1500 :
      p=700*3/100 + 300*6/100  + 500*10/100 + (m-1500)*20/100

   print "Ο μισθός του υπαλλήλου ",on, "θα μειωθεί κατά ", p, "ευρώ"
   P.append(p)

   NE.append(m-p)

   S=S+m
   S1=S1 + m-p

print " Κέρδος εταιρείας ", S- S1

N=len(NE)

for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if NE[j]>NE[j-1]:
         NE[j],NE[j-1]=NE[j-1],NE[j]
         P[j],P[j-1]=P[j-1],P[j]
         M[j],M[j-1]=M[j-1],M[j]
         F[j],F[j-1]=F[j-1],F[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
c=1
for x in range (10):
   if c<=10 :
      if F[x]=="A":
         print ON[x]
         c=c+1
```

ΑΣΚΗΣΗ 2

```
P=250
c1=0
c2=0
S=0

T=raw_input('Δώσε τύπο οχήματος')
if T=="A":
   K=1
if T=="F":
   K=2

while K <=P :
   P=P-K
   X=input('Δώσε ώρες')

   if T=="A":
      c1=c1+1
      if X==1:
         Xr=X*2.0
      if X==2:
         Xr=1*2.0 + (X-1)*1.80
      if X>2:
         Xr=1*2.0 + 1*1.80 + (X-2)*1.50
   if T=="F":
      c2=c2+1
      if X>=1 and X<=2:
         Xr=X*2.20
      if X>=3 and X<=5:
         Xr=2*2.20 + (X-2)*2.00
      if X>5:
         Xr=2*2.20 + 3*2.00 + (X-5)*1.80

   print "θα πληρώσει ", Xr, "ευρώ"

   S=S+Xr

   T=raw_input('Δώσε τύπο οχήματος')
   if T=="A":
      K=1
   if T=="F":
      K=2

   while P - K < 0:
      T=str(raw_input('Δώσε ΣΩΣΤΟ τύπο οχήματος'))
      if T=="A":
         K=1
      if T=="F":
         K=2

print "Συνολικά αυτοκίνητα ", c1
print "Συνολικά φορτηγά ", c2
print "Σύνολο χρημάτων που θα εισπράξει ", S
```

1) Ένας όμιλος επιχειρήσεων επιθυμεί να αναπτύξει ένα λογισμικό το οποίο θα χρησιμοποιηθεί από το τμήμα διαφήμισης του ομίλου για παραγωγή εξατομικευμένων συστάσεων ώστε να υλοποιηθούν νέα προϊόντα. Το λογισμικό θα αξιοποιηθεί σε τηλεφωνική έρευνα σε συγκεκριμένο δείγμα πολιτών. Σας ανατίθεται η διαδικασία ανάπτυξης του παραπάνω έργου. Για το σκοπό αυτό να δημιουργήσετε πρόγραμμα σε Python το οποίο:

[Δ1]. Θα δέχεται ως είσοδο 10 ερωτήσεις και θα τις αποθηκεύει σε κατάλληλη λίστα ΕΡ. Για παράδειγμα μια πιθανή ερώτηση θα μπορεί να είναι «Αγοράζετε επιτραπέζια παιχνίδια;» η οποία αποθηκεύεται ως πρόταση σε κελί της παραπάνω λίστας.

[Δ2]. Θα δέχεται ως είσοδο το πλήθος του δείγματος πολιτών για το οποίο θα γίνει η έρευνα.

[Δ3]. Για κάθε έναν από τους συμμετέχοντες στην έρευνα θα δίνεται ως είσοδος οι πιθανές του απαντήσεις σε κάθε μία από τις 10 ερωτήσεις οι οποίες μπορεί να είναι ΚΑΘΟΛΟΥ, ΑΡΚΕΤΑ, ΠΟΛΥ και θα αποθηκεύονται σε κατάλληλη λίστα.

[Δ4]. Θα δημιουργούν τρείς λίστες ΑΚ, ΑΑ και ΑΡ οι οποίες θα περιέχουν για κάθε ερώτηση το πλήθος κάθε πιθανής απάντησης. Δηλαδή στην λίστα ΑΚ θα περιέχεται το πλήθος όσων απάντησαν ΚΑΘΟΛΟΥ για τις 10 ερωτήσεις, στην ΑΑ το πλήθος όσων απάντησαν ΑΡΚΕΤΑ και στην ΑΡ το πλήθος όσων απάντησαν ΠΟΛΥ.

[Δ5]. με την βοήθεια συνάρτησης να εκτυπώνεται η κάθε ερώτηση και στη συνέχεια οι πιθανές απαντήσεις ΚΑΘΟΛΟΥ, ΑΡΚΕΤΑ, ΠΟΛΥ συνοδευόμενες από το πόσες φορές δόθηκαν ως απάντηση. Για παράδειγμα αν στην πρώτη ερώτηση έχουμε τις παρακάτω απαντήσεις ΚΑΘΟΛΟΥ:120 ΑΡΚΕΤΑ:91 ΠΟΛΥ:200 να εκτυπώνεται στην οθόνη:

Αγοράζετε επιτραπέζια παιχνίδια; ΠΟΛΥ:200 ΚΑΘΟΛΟΥ:120 ΑΡΚΕΤΑ:91

2) Σε ένα ηλεκτρονικό κατάστημα αθλητικών ειδών γίνονται προσφορές στους πελάτες ανάλογα με το είδος και τον αριθμό των ειδών που επιθυμούν να αγοράσουν. Για την αγορά ρούχων υπάρχει έκπτωση 25%, εάν κάποιος πελάτης αγοράσει τουλάχιστον 3 τεμάχια. Για την αγορά παπουτσιών, υπάρχει έκπτωση 10% και δώρο ένα ζευγάρι αθλητικές κάλτσες για κάθε ζευγάρι παπουτσιών ανεξαρτήτως των τεμαχίων που θα αγοράσει ο πελάτης. Εάν ο πελάτης επιθυμεί να αγοράσει οποιοδήποτε άλλο αθλητικό είδος από το κατάστημα λαμβάνει έκπτωση 15% για αυτό εάν συνοδεύεται από την ειδική ένδειξη ΕΚ1 και 20% εάν συνοδεύεται από την ειδική ένδειξη ΕΚ2. Δεν υπάρχει περίπτωση κάποιο από τα άλλα είδη να μη συνοδεύεται από ειδική ένδειξη ΕΚ1 ή ΕΚ2.

Το κατάστημα χρεώνει επιπλέον τον πελάτη με κάποια μεταφορικά έξοδα ανάλογα με το βάρος των τεμαχίων που αγόρασε. Συγκεκριμένα κάθε γραμμάριο κοστίζει 0,01 ευρώ για τη μεταφορά. Εάν όμως ο πελάτης αγοράσει πάνω από 10 τεμάχια το κατάστημα χαρίζει στον πελάτη τα μεταφορικά έξοδα για τα 10 πρώτα. Αν για παράδειγμα αγοράσει 12 τεμάχια θα χρεωθεί τα μεταφορικά έξοδα μόνο για το βάρος του 11ου και του 12ου τεμαχίου. Να κατασκευάσετε αλγόριθμο ο οποίος:

α) θα διαβάζει την τιμή, το βάρος (σε γραμμάρια) και το είδος κάθε προϊόντος που θέλει να προσθέσει στο ηλεκτρονικό καλάθι των αγορών του ο πελάτης. Οι αποδεκτές τιμές για το είδος του προϊόντος είναι R για τα ρούχα, P για τα παπούτσια και A για τα άλλα είδη του καταστήματος. Η επαναληπτική είσοδος στοιχείων να σταματάει όταν δοθεί ο χαρακτήρας Χ από τον πελάτη. Σε περίπτωση που δοθεί ως τιμή εισόδου ο χαρακτήρας Α, ο αλγόριθμος να διαβάζει και την ειδική ένδειξη για την έκπτωση

β) θα εμφανίζει το μήνυμα: Καλάθι αγορών άδειο και θα τερματίζει τη διαδικασία εάν ο πελάτης δεν αγοράσει κανένα προϊόν

Εάν ο πελάτης αγοράσει ένα η περισσότερα προϊόντα, θα υπολογίζει και θα εμφανίζει:

Α) πόσα ζευγάρια κάλτσες πήρε ως δώρο από το κατάστημα

Β) πόσα είναι τα μεταφορικά έξοδα κάθε φορά που προστίθεται ένα νέο προϊόν στο καλάθι, κάθε φορά δηλαδή που ο πελάτης πραγματοποιεί νέα αγορά.

Γ) το ποσοστό των άλλων ειδών που αγόρασε εκτός των παπουτσιών

Δ) πόσα είναι τα συνολικά χρήματα, μαζί με τα μεταφορικά έξοδα, που ξόδεψε για τις αγορές του

Σημείωση: Δεν απαιτείται έλεγχος εγκυρότητας για τις τιμές εισόδου

3) Στο ράλλυ Ακρόπολις 2013 συμμετέχουν 76 οδηγοί σε 18 ειδικές διαδρομές. Η επιτροπή των αγώνων καταγράφει το χρόνο κάθε οδηγού σε κάθε ειδική διαδρομή καθώς και τους βαθμούς ποινής οι οποίοι προκύπτουν από πιθανές παραβάσεις κανονισμών και «αυξάνουν» τον πραγματικό χρόνο τερματισμού κάθε οδηγού. Οι κατασκευάστριες εταιρίες αυτοκινήτων που συμμετέχουν στον αγώνα είναι 38, δηλαδή κάθε μία συμμετέχει στο ράλλυ με 2 οδηγούς.

Να κατασκευάσετε πρόγραμμα το οποίο:

Α) να διαβάζει το όνομα καθενός από τους 76 οδηγούς καθώς και το όνομα της κατασκευάστριας εταιρίας που του παρέχει το όχημα, στις λίστες ΟΝ και ΚΑΤ αντίστοιχα. Επίσης να διαβάζει το χρόνο σε δευτερόλεπτα κάθε οδηγού σε κάθε ειδική διαδρομή εξασφαλίζοντας πως θα είναι θετικός αριθμός καθώς και τους βαθμούς ποινής κάθε οδηγού σε κάθε ειδική διαδρομή εξασφαλίζοντας πως είναι μη αρνητικός αριθμός.

Β) να «αυξάνει» το χρόνο κάθε οδηγού σε κάθε ειδική διαδρομή ανάλογα με τους βαθμούς ποινής που έχει πάρει στη διαδρομή σύμφωνα με τον παρακάτω πίνακα

Βαθμοί ποινής	**Αύξηση χρόνου**
1-10	0.5 δευτερόλεπτα ανά βαθμό
11-20	1 δευτερόλεπτο ανά βαθμό
Πάνω από 20	1.5 δευτερόλεπτο ανά βαθμό

Σημείωση: Η αύξηση του χρόνου να πραγματοποιηθεί κλιμακωτά

Γ) να δημιουργεί λίστα SUM με το συνολικό χρόνο κάθε οδηγού σε όλη τη διάρκεια του ράλλυ. Στη συνέχεια να εμφανίζει το όνομα του οδηγού που κέρδισε στον αγώνα καθώς και το όνομα της κατασκευάστριας εταιρίας με την οποία αγωνίστηκε. Επίσης να εμφανίζει το όνομα και το χρόνο που πέτυχε ο δεύτερος οδηγός της ίδιας κατασκευάστριας εταιρίας.

Δ) να εμφανίζει την κατάταξη από τον καλύτερο οδηγό προς το χειρότερο, μαζί με το χρόνο που πέτυχε σε αυτήν και την εταιρεία του. Για το σκοπό αυτό να καλεί τη διαδικασία ΕΙΔΙΚΗ, η οποία περιγράφεται στο επόμενο ερώτημα.

Ε) να κατασκευάσετε τη διαδικασία ΕΙΔΙΚΗ, η οποία θα δέχεται την λίστα των χρόνων, την λίστα με τα ονόματα των οδηγών και την λίστα με τα ονόματα των εταιριών και θα επιστρέφει στο κύριο πρόγραμμα τις λίστες αυτές ταξινομημένες ώστε αυτό να εμφανίζει τα αποτελέσματα.

ΑΣΚΗΣΗ 1

```
def  info(SUM, EP):
   for y in range(10):
      c1=0
      c2=0
      c3=0
      for x in range(len(SUM)):
         if SUM [x] [y] ==1:
            c1+=1
         if SUM [x] [y] ==2:
            c2+=1
         if SUM [x] [y] ==3:
            c3+=1
      print  EP[y], ": ΚΑΘΟΛΟΥ ",c1, "ΑΡΚΕΤΑ", c2, "ΠΟΛΥ", c3

EP=[]
AK=[]
AA=[]
AP=[]
SUM=[]

for x  in range(10):
   ep=input('Δώσε την ερώτηση')
   EP.append(ep)

N=input('Δωσε το δείγμα των πολιτών')

for x  in range(N):
   c1=0
   c2=0
   c3=0
   T=[]
   for y in range(10):

      print EP[y]
      apadisi=input('Δώσε απάντηση 1 ΚΑΘΟΛΟΥ 2 ΑΡΚΕΤΑ 3 ΠΟΛΥ')
      T.append(apadisi)
      if apadisi==1:
         c1+=1
      if apadisi==2:
         c2+=1
      if apadisi==3:
         c3+=1

   SUM.append(T)
   AK.append(c1)
   AA.append(c2)
   AP.append(c3)

info(SUM, EP)
```

ΑΣΚΗΣΗ 2

```
pr=0
k=0
k1=0
ME=0
F=False
S=0

E=raw_input('Δώσε είδος')
while E not in ["R", "P", "A", "X"] :
   E=raw_input('Δώσε ΣΩΣΤΟ είδος')

while E!="X":
   pr+=1

   T=input('Δώσε τιμή')
   B=input('Δώσε βάρος')

   if E=="P":
      k=k+1
      Ek=10
   elif E=="A":
      EE=raw_input("Δώσε ειδική ένδειξη ΕΚ1 ή ΕΚ2")
      k1=k1+1
      if EE=="EK1":
         Ek=15
      if EE=="EK2":
         Ek=20
   else:
      k1=k1+1
      Ek=25

   if pr<=10:
      ME = ME + B*0.01
   else:
      if F==False:
         ME=0
         F=True
      ME = ME + B*0.01
   print "Μεταφορικά έξοδα ", ME

   S=S + T - T*Ek/100 + ME

   E=raw_input('Δώσε είδος')
   while E not in ["R", "P", "A", "X"] :
      E=raw_input('Δώσε ΣΩΣΤΟ είδος')

if pr==0:
   print "Καλάθι αγορών άδειο "
else:
   print "Ζευγάρια κάλτσες δώρο :", k
   print "ποσοστό των άλλων ειδών εκτός παπουτσιών", 100*k1/pr
   print "Συνολικά χρήματα", S
```

ΑΣΚΗΣΗ 3

```
def EIDIKI(A, B, C):
   N=len(A)
   for i in range(1,N,1):
      for j in range (N-1,i-1,-1):
         if C[j] < C[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
            C[j],C[j-1]=C[j-1],C[j]

ON=[]
KAT=[]
SUM=[]

for x  in range(76):
   on=str(raw_input('Δώσε ονομα'))
   kat=str(raw_input('Δώσε εταιρία'))
   On.append(on)
   KAT.append(kat)

   for y  in range(18):
      XR=input('Δώσε χρόνο')
      while XR<0:
         XR=input('Δώσε χρόνο')
      p=input('Δώσε ποινή')

      if p>=1 and p<=10:
         XR=XR + p*0.5
      if p>=11 and p<=20:
         XR=XR + 10*0.5 + (p-10)*1
      if p>20:
         XR=XR + 10*0.5 + 10*1 + (p-20)*1.5

   SUM.append(XR)

min1=SUM[0]
for x in range (1,76):
   if SUM[x]< min1:
      min1=SUM[x]
      Z=x

print ON[Z], KAT[Z]

for x in range (76):
   if KAT[x]==KAT[Z] and ON[x] != ON[Z]:
      print "Ο 2ος Οδηγός ", ON[x], SUM[x]

EIDIKI(ON, KAT, SUM)

for x in range (76):
   print ON[x], KAT[x], SUM[x]
```

1) Σε ένα σύμπλεγμα νησιών υπάρχουν 5 νησιά κάθε ένα από τα οποία έχει έναν αναγνωριστικό κωδικό από 1341 μέχρι και το 1345. Από το συγκεκριμένο σύμπλεγμα λαμβάνουν μέρος σε διαγωνισμό πληροφορικής 140 μαθητές οι οποίοι προέρχονται από οποιοδήποτε νησί (τουλάχιστον ένας από κάθε νησί).

Να γραφεί πρόγραμμα το οποίο θα διαβάζει για κάθε μαθητή:

Α. Τον κωδικό του νησιού από το οποίο προέρχεται και να τον αποθηκεύσει σε κατάλληλη λίστα. Να θεωρήσετε ότι ο χρήστης πληκτρολογεί έγκυρες τιμές με τον εξής τρόπο για το νησί με κωδικό 1341 δίνει 1, για το νησί με κωδικό 1342 δίνει 2 κ.ο.κ. .

Β. Τον βαθμό που έλαβε στον διαγωνισμό πληροφορικής και να τον αποθηκεύει σε κατάλληλη λίστα. Να θεωρήσετε πως ο χρήστης πληκτρολογεί τιμές από το 1 έως και το 5.

Στη συνέχεια το πρόγραμμα θα πρέπει να κατασκευάζει την λίστα των κωδικών των νησιών. Η λίστα θα αποτελείται από 5 θέσεις.

Να υπολογιστεί και να εκτυπωθεί:

Α.Ο μέσος όρος όλων των μαθητών στον διαγωνισμό πληροφορικής.

Β. Πόσοι μαθητές έγραψαν πάνω από 3 στον διαγωνισμό.

Γ. Πόσοι μαθητές από κάθε νησί συμμετείχαν στον διαγωνισμό.

Δ. Τον μέσο όρο βαθμολογίας κάθε νησιού.

Ε. Τον κωδικό του νησιού με τον μεγαλύτερο μέσο όρο βαθμολογίας.

2) Σε ένα σύνολο 100 ατόμων τοποθετείται προληπτικά μια συσκευή καταγραφής καρδιακής λειτουργίας προκειμένου να παρακολουθήσει σε μια ημέρα βασικά χαρακτηριστικά στοιχεία της καρδιάς. Η καταγραφή των στοιχείων γίνεται με ειδικούς αισθητήρες οι οποίοι τοποθετούνται σε 3 διαφορετικά σημεία του σώματος. Ο πρώτος αισθητήρας καταγράφει παλμούς, ο δεύτερος θερμοκρασία σώματος και ο τρίτος πίεση. Οι μετρήσεις λαμβάνονται ***ανά ώρα σε χρόνο μιας ημέρας.***

Να γραφεί αλγόριθμος ο οποίος:

Α. Για κάθε άτομο:

i. Διαβάζει την ηλικία και το φύλο του (Α-ΑΝΔΡΑΣ, Γ-ΓΥΝΑΙΚΑ).

ii. Για κάθε ώρα και για μια ημέρα να διαβάζει τις μετρήσεις από τους τρεις αισθητήρες και να τους αποθηκεύει σε κατάλληλες τρεις λίστες P, PA, T .

iii. Υπολογίζει και εκτυπώνει με την βοήθεια τριών κατάλληλων συναρτήσεων PIESI, PALMOI, THERM:

1.Τον μέσο όρο παλμών του κάθε ατόμου (Μέσος όρος ημέρας).

2. Την μέγιστη τιμή πίεσης του κάθε ατόμου (Μέγιστη τιμή ημέρας).

3.Πόσες φορές η θερμοκρασία σώματος ξεπέρασε τους 36.7 βαθμούς Κελσίου (Μέσα στην ημέρα).

4.Πόσες φορές οι παλμοί ξεπέρασαν το όριο των 80 παλμών. Επίσης να γίνει και εκτύπωση της ώρας στην οποία παρατηρήθηκε το παραπάνω.

5.Το ποσοστό μετρήσεων πίεσης με τιμή 10.

Β. Να υπολογιστεί και να εκτυπωθεί ο μέσος όρος ηλικίας των ατόμων, το πλήθος των γυναικών και το πλήθος των ανδρών που συμμετείχαν στην διαδικασία.

Γ. Να υπολογιστεί και να εκτυπωθεί η μέγιστη τιμή πίεσης και ποιο άτομο από τα 100 την παρουσίασε. Σε περίπτωση ύπαρξης πολλών ατόμων με ίδια τιμή μέγιστης πίεσης να εμφανιστεί ο τελευταίος από αυτούς.

Δεν βλάπτει τον αλγόριθμο μόλις ολοκληρώσω τις ενέργειες για τον πρώτο ασθενή να χρησιμοποιήσω τις ίδιες λίστες και για τον επόμενο αφού πρώτα τις αδειάσω.

3) Σε ένα σχολείο στην Αττική ο καθηγητής προγραμματισμού επιθυμεί να παρακολουθήσει την εξέλιξη των μαθητών του. Για το λόγο αυτό σχεδίασε μια εφαρμογή λογισμικού. Καλείστε να δημιουργήσετε τον αλγόριθμο για την εφαρμογή αυτή.

Ο αλγόριθμος θα πρέπει για κάθε έναν από τους 234 μαθητές του σχολείου:

Α. Να διαβάζει τους βαθμούς του σε δέκα ολιγόλεπτα διαγωνίσματα και θα τους αποθηκεύει σε κατάλληλη λίστα με όνομα Τ. Να θεωρήσετε πως όλοι οι μαθητές συμμετέχουν σε όλα τα διαγωνίσματα.

Β. Να διαβάζει την ημερομηνία του κάθε διαγωνίσματος και να την αποθηκεύει σε λίστα ΗΜ.

Γ. Εκτυπώνει τις ημερομηνίες που ο μαθητής έγραψε βαθμό μεγαλύτερο από 17.

Δ. Υπολογίζει και εκτυπώνει τον μέσο όρο στα τεστ κάθε μαθητή.

Ε. Υπολογίζει και εκτυπώνει την ημερομηνία στην οποία ο μαθητής έγραψε μέγιστο βαθμό σε τεστ. Ενδέχεται να είναι πολλές.

ΣΤ. Εκτυπώνει κατά φθίνουσα σειρά ως προς την βαθμολογία τους βαθμούς στα τεστ συνοδευόμενα από την ημερομηνία.

Παρατήρηση: Οι δύο λίστες που θα φτιάξετε θα γεμίσουν με δεδομένα 234 φορές. Δεν βλάπτει τον αλγόριθμο μόλις ολοκληρώσω τις ενέργειες για τον πρώτο μαθητή να χρησιμοποιήσω τις ίδιες λίστες και για τον επόμενο. Προσοχή στην σειρά με την οποία θα εκτελέσετε τα ερωτήματα.

4) Σε έναν αγώνα αυτοκινήτων δήλωσαν συμμετοχή 8 οδηγοί οι οποίοι με τα αυτοκίνητα τους έτρεξαν σε 14 ειδικές διαδρομές σε 14 πόλεις της Ελλάδας.

Να γραφτεί πρόγραμμα το οποίο :

Α) θα διαβάζει και θα αποθηκεύει σε κατάλληλη λίστα τα ονόματα των 14 πόλεων της Ελλάδας

Β) Για κάθε οδηγό θα διαβάζει το όνομα του και την επίδοση του σε κάθε αγώνα

Γ) Θα εκτυπώνει το όνομα της πόλης όπου ο κάθε οδηγός έκανε τον ταχύτερο αγώνα.

Δ) Θα εκτυπώνει το όνομα κάθε οδηγού στον οποίο παρατηρήθηκε συνεχής βελτίωση της επίδοσης του από αγώνα σε αγώνα. Ίδια επίδοση μεταξύ δύο συνεχόμενων αγώνων δεν θεωρείτε βελτίωση.

Ε) Να εκτυπωθεί το όνομα της πόλης η οποία συγκέντρωσε τις περισσότερες καλύτερες επιδόσεις των οδηγών

ΑΣΚΗΣΗ 1

```
N=range(1341, 1346)

K=[]
B=[]

for x in range(140):
   k=input('Δωσε κωδικό νησιού')
   K.append(k)

   b=input('Δωσε βαθμό')
   while b<1 or b>5:
      b=input('Δωσε σωστό βαθμό')
   B.append(b)

S=0.0
c=0
for x in B:
   S=S+x

   if x >3:
      c=c+1

print "Ο Μέσος Όρος βαθμολογιών", S/140

print "Εγραψαν πάνω απο 3 ", c

c1=c2=c3=c4=c5=0

for x in K :
   if x==1:
      c1+=1
   if x==2:
      c2+=1
   if x==3:
      c3+=1
   if x==4:
      c4+=1
   if x==5:
      c5+=1

print "Συμμετείχαν απο το νησί ",N[0], c1, "μαθητές"
print "Συμμετείχαν απο το νησί ",N[1], c2, "μαθητές"
print "Συμμετείχαν απο το νησί ",N[2], c3, "μαθητές"
print "Συμμετείχαν απο το νησί ",N[3], c4, "μαθητές"
print "Συμμετείχαν απο το νησί ",N[4], c5, "μαθητές"
```

```
max1=0
for x  in range(1, 6):
   for y  in range(140):
      z=0
      S=0.0
      if K[y]==x:

         S=S+B[y]
         z=z+1
   print"Ο Μέσος όρος του ",N[x-1], "ειναι",S/z

   if S/z > max1:
      max1=S/z
      k1= x-1

print "Νησί με μεγαλύτερο μέσο όρο", N[K1]
```

ΑΣΚΗΣΗ 2

```
def  PALMOI(PA):
   c=0
   S=0.0
   A=[]
   for x  in range(len(PA)):
      S=S+PA[x]

   if  PA[x] >80:
      c=c+1
      A.append(x)

   print "Μέσος Όρος παλμών", S/24
   print "ξεπερασε τους 80 παλμούς ", c, "φορές"
   for x in A:
      print "Ωρα που ξεπερασε τους 80 παλμούς ", x


def PIESI(P):
   max1=P[0]
   for x  in range(1,len(P)):
      if P[x]>max1:
         max1=P[x]

   print "Μεγιστη πίεση: ",max1
   MAXP.append(max1)

   c=0
   for x  in P:
      if x==10:
         c=c+1

   print "Ποσοστό με πίεση 10 ",100*c/24


def THERM(T):
   c=0
   for x  in T:
      if  x > 36.7:
         c=c+1
   print "Ξεπερασε τους 36.7", c, "φορές"

H=[]
F=[]
MAXP=[]
for x  in range(100):
   h=input("Δωσε ηλικια")
   f=str(raw_input("Δωσε το φύλο"))
   H.append(h)
   F.append(f)

   P=[]
   PA=[]
```

```
   T=[]
   for  y  in range(24):
      p=input('Δωσε πίεση')
      P.append(p)
      pa=input('Δωσε παλμούς')
      PA.append(pa)
      t=input('Δωσε θερμοκρασία')
      T.append(t)

   PALMOI(PA)
   PIESI(P)
   THERM(T)



   while P!=[]:
      P.pop()
      PA.pop()
      T.pop()

S=0.0
for x in H:
   S=S+x

print "Μέσος Όρος ηλικιών", S/100

c=0
for x in F:
   if x=="A":
      c=c+1
print "Πλήθος ανδρών", c, "και πλήθος γυναικών ", 100-c

max1=MAXP[0]
for x in range(1,100):
   if MAXP[x]>max1:
      max1=MAXP[x]
      Z=x

print "Την μεγιστη πίεση ",max1, "εμφάνισε ο", Z
```

ΑΣΚΗΣΗ 3

```
def BSort(A,B):
   N=len(B)
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if B[j] > B[j-1]:
            B[j], B[j-1]= B[j-1], B[j]
            A[j], A[j-1]= A[j-1], A[j]

for x  in range(234):
   T=[]
   HM=[]
   for  y  in range(10):
      v=input('Δώσε Βαθμό')
      T.append(v)
      hm=str(raw_input('Δώσε ημορομηνία'))
      HM.append(hm)

   S=0.0
   for i  in range(len(T)):
      if T[i] > 17:
         print HM[i]

      S=S+T[i]

   print "Μέσος Όρος των τεστ: ", S/len(T)

   max1=T[0]
   for i in range(1,len(T)):
      if T[i]> max1:
         max1=T[i]

   for i in range(len(T)):
      if T[i]==max1 :
         print HM[i]

   BSort(HM,T)

   for i in range(len(T)):
      print T[i], HM[i]

   while T !=[]:
      T.pop()
      HM.pop()
```

ΑΣΚΗΣΗ 4

```
def   count_My(A, p):
   c=0
   for x  in A:
      if x==p:
         c=c+1
   return c

N=[]
for  x  in range(14):
   n=str(raw_input('Δώσε όνομα Πόλης'))
   N.append(n)

P=[]
for x  in range(8):
   on=raw_input('Δώσε όνομα Οδηγού')
   E=[]
   for  y  in range(14):
      ep=input('Δώσε επίδοση')
      E.append(ep)

   min1=E[0]
   for i  in range(1,len(E)):
      if E[i]<min1:
         min1=E[i]
         Z=i

   print "Πόλη με ταχύτερο αγώνα :", N[Z], ‘ του οδηγού ‘, on
   P.append(N[Z])

   F=False
   for i  in range(len(E)-1):
      if E[i] <= E[i+1] :
         F=True
   if F==False :
      print on

   while E!=[]:
      E.pop()

max1=0
for x  in N:
   if count_My(P, x) > max1:
      max1=count_My(P, x)
      Z=x

print "Πόλη με τις καλύτερες επιδόσεις : ", Z
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ.

ΘΕΩΡΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΚΕΦ 3.

Βασικά στοιχεία γλώσσας προγραμματισμού

Επιλέξτε τον σωστό τύπο δεδομένων στον οποίο ανήκουν οι παρακάτω τιμές

1) Ταιριάξτε την αριστερή με την δεξιά στήλη:

1) True	Α) Ακέραιος
2) 'TEST'	Β) Λογικός
3) 23.0	Γ) Κινητής Υποδιαστολής
4) "120.5"	Δ) Συμβολοσειρά ή Αλφαριθμητικό
5) "False"	
6) 13.523	
7) 28.4E-5	
8) *"23"	
9) 14/5	
10) 5/2.0	

2) Να υπολογίσετε ποιο θα είναι στην Python το αποτέλεσμα των παρακάτω πράξεων (Να γράφετε το κόμμα με τελεία)

Α) 5+8%2=

Β) 6/2+1=

Γ) 5/2=

Δ) 5/2.0=

Ε) 5-2/2=

Ζ) 19%(5+1)=

Η) (10-5)*2=

Θ) 5**2+3=

Ι) 3+4/2**2=

Κ) 10+7/3*2=

3) Να χαρακτηρίσετε καθεμιά από τις ακόλουθες λογικές εκφράσεις ως True ή False αν A=5 και B=7

Α) A<=9

Β) B>7

Γ) A!=B

Δ) B==A

Ε) A>=5

Ζ) A>3 and B<3

Η) not (A!=4 or B==8)

Θ) A>=2 and B<=9

4) Πώς θα γράψω στην python: Ο βαθμός να είναι από 0 μέχρι και 20

Α) V > 0 and V<20

Β) V >= 0 and V<=20

Γ) V <= 0 and V>=20

Δ) V < 0 and V>20

5) Πώς θα γράψω στην python: Όλοι οι αριθμοί που είναι κάτω από 0 και οι αριθμοί που είναι πάνω από 100

Α) V < 0 and V>=100

Β) V <= 0 or V>100

Γ) V < 0 or V>100

Δ) V < 0 and V>100

6) Ταιριάξτε τις δύο στήλες

Α) float(x)	1. Επιστρέφει την δύναμη ενός αριθμού (έναν αριθμό υψωμένο σε έναν άλλο)
Β) int(x)	2. Παίρνει σαν είσοδο έναν αριθμό (ή μία συμβολοσειρά) και επιστέφει μία συμβολοσειρά
Γ) str(x)	3. Επιστρέφει τη ρίζα ενός αριθμού
Δ) pi	4. Βιβλιοθήκη μαθηματικών
Ε) abs(x)	5. Παίρνει σαν είσοδο έναν αριθμό (ή μία συμβολοσειρά) και επιστρέφει τον αριθμό ως κινητής υποδιαστολής
Ζ) pow(x,y)	6. Επιστρέφει την απόλυτη τιμή ενός αριθμού
Η) divmod(x,y)	7. Επιστρέφει το πηλίκο της ακέραιας διαίρεσης και το ακέραιο υπόλοιπο δύο αριθμών
Θ) math	8. Επιστέφει το π=3.14....
Ι) randint(x,y)	9. Επιστρέφει έναν τυχαίο ακέραιο αριθμό από το αρχικό μέχρι και το τελικό όριό της
Κ) sqrt(x)	10. Παίρνει σαν είσοδο έναν αριθμό (ή μία συμβολοσειρά) και επιστρέφει το ακέραιο μέρος

7) Ταιριάξτε τις δύο στήλες

Α) >=	1. Αριθμητικός Τελεστής
Β) and	2. Τελεστής λογικών πράξεων
Γ) **	3. Σχεσιακός (ή συγκριτικός) τελεστής
Δ) !=	
Ε) ==	
Ζ) or	
Η) not	
Θ) %	
Ι) *	
Κ) <	

8) Χαρακτηρίστε τους παρακάτω τύπους δεδομένων

Α) Πίνακας	1) Απλός Τύπος Δεδομένων
Β) Ακέραιος	2) Σύνθετος Τύπος Δεδομένων
Γ) Πραγματικός	
Δ) Ουρά	
Ε) Λογικός	
Ζ) Δένδρο	
Η) Αλφαριθμητικός	
Θ) Λίστα	
Ι) Χαρακτήρας	
Κ) Σύνολο	

ΑΠΑΝΤΗΣΕΙΣ

1)

1	Λογικός (bool)
2	Συμβολοσειρά ή αλφαριθμητικό (str)
3	Κινητής υποδιαστολής (float)
4	Συμβολοσειρά ή αλφαριθμητικό (str)
5	Συμβολοσειρά ή αλφαριθμητικό (str)
6	Κινητής υποδιαστολής (float)
7	Κινητής υποδιαστολής (float)
8	Συμβολοσειρά ή αλφαριθμητικό (str)
9	Ακέραιος (int)
10	Κινητής υποδιαστολής (float)

2)

Α	5
Β	4
Γ	2
Δ	2.5
Ε	4
Ζ	1
Η	10
Θ	28
Ι	4
Κ	14

3)

Α	True
Β	False
Γ	True
Δ	False
Ε	True
Ζ	False
Η	False
Θ	True

4) B>=0 and B<=20

5) A<0 or A>100

6)

Α	5
Β	10
Γ	2
Δ	8
Ε	6
Ζ	1
Η	7
Θ	4
Ι	9
Κ	3

7)

Α	3
Β	2
Γ	1
Δ	3
Ε	3
Ζ	2
Η	2
Θ	1
Ι	1
Κ	3

8)

Α	2
Β	1
Γ	1
Δ	2
Ε	1
Ζ	2
Η	2
Θ	2
Ι	1
Κ	2

ΘΕΩΡΙΑ ΚΕΦΑΛΑΙΟ 3

1) Στην Python δηλώνουμε ποιο τύπο δεδομένων χρησιμοποιούμε. Σ Λ

2) Ποιοι είναι οι χαρακτηριστικοί τύποι δεδομένων στην Python ;

3) Ποιοι είναι οι αριθμητικοί τύποι δεδομένων ;

4) Τι δηλώνει το σύμβολο E; Τι σημαίνει 5.3E2 ;

5) Σε πραγματικούς αριθμούς στην Python το δεκαδικό τμήμα διαχωρίζεται με κόμμα "," Σ Λ

6) Ο λογικός τύπος (boolean) έχει σκοπό την καταγραφή του αποτελέσματος _____ _____

7) Μπορούμε να ορίσουμε μια συμβολοσειρά με μονά εισαγωγικά ή με διπλά, αλλά όχι _____

8) Ποια εντολή χρησιμοποιούμε για να ελέγξουμε τον τύπο δεδομένων ;

9) Τι είναι οι τελεστές ;

10) Ποιοι είναι οι βασικότεροι τελεστές στη γλώσσα Python ;

11) Ποια είναι η ιεραρχία πράξεων στους αριθμητικούς τελεστές ;

12) Ποια είναι η προτεραιότητα (ιεραρχία) των λογικών τελεστών;

13) Αν θέλουμε να αλλάξουμε την ιεραρχία των πράξεων, μπορούμε να χρησιμοποιήσουμε _____

14) Στην Python, η διαίρεση ακεραίων αριθμών μας επιστρέφει _____

15) Ποιοι είναι οι κανόνες που πρέπει να ακολουθούμε σχετικά με το όνομα μιας μεταβλητής;

16) Όλα τα δεδομένα σε ένα πρόγραμμα Python αναπαρίστανται με αντικείμενα ή με σχέσεις μεταξύ των αντικειμένων, με κάθε αντικείμενο να έχει μια _____, έναν _____ και μία _____.

17) Στο παράδειγμα a = 125, δημιουργείται η μεταβλητή με _____ a και αναφέρεται στο αντικείμενο, με _____ 125, με ακέραιο _____ _____.

18) Τι ονομάζεται συλλογή σκουπιδιών ;

19) Ποιο σύμβολο χρησιμοποιούμε για τα σχόλια ;

20) Αναφέρετε μερικές Βασικές (ενσωματωμένες) συναρτήσεις

21) Οι βιβλιοθήκες εισάγονται με την εντολή _____

22) Η συνάρτησης για την τετραγωνική ρίζα, είναι η _____

23) Ποιοι είναι οι Απλοί και ποιοι οι σύνθετοι τύποι δεδομένων;

ΑΠΑΝΤΗΣΕΙΣ

Οι τύποι δεδομένων προσδιορίζουν τον τρόπο παράστασης των δεδομένων εσωτερικά στον υπολογιστή, καθώς και το είδος της επεξεργασίας τους από αυτόν. Στην Python δε δηλώνουμε τύπο δεδομένων. σελ 29

Ποιοι είναι οι χαρακτηριστικοί τύποι δεδομένων στην Python ; σελ 29

Οι χαρακτηριστικοί τύποι δεδομένων στην Python είναι ο αριθμητικός, ο λογικός (boolean) και οι συμβολοσειρές ή αλφαριθμητικά (strings).

Ποιοι είναι οι αριθμιτικοί τύποι δεδομένων ; σελ 29

Οι αριθμοί στην Python είναι κυρίως τριών τύπων:

α) ακέραιοι, β) αριθμοί κινητής υποδιαστολής γ) μιγαδικοί αριθμοί.

Το σύμβολο Ε δηλώνει δύναμη του 10 π.χ. 28.2E-5 σημαίνει $28.2 * 10^{-5}$ σελ 29

Το δεκαδικό τμήμα διαχωρίζεται με το χαρακτήρα τελεία “.” και όχι το κόμμα “,”. σελ 29

Ο λογικός τύπος (boolean) έχει σκοπό την καταγραφή του αποτελέσματος ενός ελέγχου. σελ 30

Μπορούμε να ορίσουμε μια συμβολοσειρά με μονά εισαγωγικά ή με διπλά, αλλά όχι ανάμικτα. σελ 30

Για να ελέγξουμε τον τύπο δεδομένων χρησιμοποιούμε την εντολή type (). σελ 30

Τι είναι οι τελεστές ; σελ 30

Οι τελεστές (operators) είναι σύμβολα ή λέξεις για τη δημιουργία αριθμητικών και λογικών εκφράσεων.

Ποιοι είναι οι βασικότεροι τελεστές στη γλώσσα Python ; σελ 31

Οι βασικότεροι τελεστές στη γλώσσα Python είναι:

α) Αριθμητικοί, β) Σχεσιακοί (ή συγκριτικοί) γ) Λογικών πράξεων

Ποια είναι η ιεραρχία πράξεων στους αριθμητικούς τελεστές ; σελ 31

1. Ύψωση σε δύναμη.

2. Πολλαπλασιασμός, διαίρεση, υπόλοιπο ακεραίας διαίρεσης.

3. Πρόσθεση, αφαίρεση.

Αν θέλουμε να αλλάξουμε την ιεραρχία των πράξεων, μπορούμε να χρησιμοποιήσουμε παρενθέσεις.

Η προτεραιότητα των λογικών τελεστών είναι not, and, or με αυτή τη σειρά. σελ 32

Στην Python, η διαίρεση ακεραίων αριθμών μας επιστρέφει το ακέραιο πηλίκο π.χ. 45 / 10 = 4 σελ 32

Η διαίρεση αριθμών κινητής υποδιαστολής, μας επιστρέφει το πηλίκο, ως αριθμό κινητής υποδιαστολής 45.0 / 10 = 4.5

Για τη χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της, ενώ μπορεί να εκχωρήσουμε διαφορετικούς τύπους τιμών σε μια μεταβλητή κατά τη διάρκεια ενός προγράμματος, όπως ακέραιες τιμές, κινητής υποδιαστολής και συμβολοσειρές. σελ 33

Ποιοι είναι οι κανόνες που πρέπει να ακολουθούμε σχετικά με το όνομα μιας μεταβλητής; σελ 33

α) δεν επιτρέπεται να ξεκινά το όνομα μιας μεταβλητής με αριθμό β) το όνομα που θα δώσουμε δεν πρέπει να είναι όμοιο με κάποιο όνομα ενσωματωμένης συνάρτησης ή εντολής, γ) δεν επιτρέπονται τα κενά δ) δεν επιτρέπονται ειδικοί χαρακτήρες όπως το $, @, οι αριθμητικοί και οι σχεσιακοί τελεστές ε) Υπάρχει διάκριση πεζών και κεφαλαίων γραμμάτων

Όλα τα δεδομένα σε ένα πρόγραμμα Python αναπαρίστανται με αντικείμενα ή με σχέσεις μεταξύ των αντικειμένων, με κάθε αντικείμενο να έχει μια ταυτότητα (identity), έναν τύπο και μία τιμή. Για παράδειγμα, το 12 είναι ένα αντικείμενο με τιμή 12, τύπου int (ακέραιος). σελ 34-35

Στο παράδειγμα a = 125, δημιουργείται η μεταβλητή με όνομα a και αναφέρεται στο αντικείμενο, με τιμή 125, με ακέραιο τύπο δεδομένων. Το σύμβολο "=" δημιουργεί ένα είδος δεσίματος μεταξύ του αντικειμένου 125 και της μεταβλητής με όνομα a. Ο τύπος της μεταβλητής είναι και αυτός ακέραιος, αφού αναφέρεται σε αντικείμενο με ακέραιο τύπο δεδομένων.

Όταν θέσουμε στη συνέχεια a = 250.0, η μεταβλητή a παύει πλέον να δείχνει το αντικείμενο με τιμή 125 και δημιουργείται ένα νέο “δέσιμο”, ώστε να αναφέρεται στο αντικείμενο με τιμή 250.0 (κινητής υποδιαστολής τύπος δεδομένων). Ο τύπος δεδομένων της μεταβλητής a τώρα έχει αλλάξει και είναι κινητής υποδιαστολής, όμοιος με τον τύπο του νέου αντικειμένου στο οποίο αναφέρεται. σελ 35

Τι ονομάζεται συλλογή σκουπιδιών ; σελ 35

Η Python παρακολουθεί όλες τις τιμές και τις διαγράφει όταν πάψουν να υπάρχουν μεταβλητές που να αναφέρονται σε αυτές. Η διαδικασία αυτή ονομάζεται συλλογή σκουπιδιών (garbage collection).

Στην Python, τα σχόλια εισάγονται θέτοντας μπροστά από αυτά το σύμβολο # σελ 38

Βασικές (ενσωματωμένες) συναρτήσεις : α) float(), β) int(), γ) str(), δ) abs(), ε) pow(a,b), στ) divmod(x,y) σελ 39

Οι βιβλιοθήκες εισάγονται με την εντολή import. σελ 40

Η συνάρτησης για την τετραγωνική ρίζα, είναι η **sqrt()**. σελ 40

```
import math

riza = math.sqrt(2)
```

Σελ 42 - 43

Τύποι Δεδομένων :

α) Απλοί τύποι δεδομένων β) Σύνθετος τύπος δεδομένων

Ποιοι είναι οι Απλοί και ποιοι οι σύνθετοι τύποι δεδομένων;

Απλοί: α) Ακέραιος β) Πραγματικός γ) Χαρακτήρας δ) Λογικός ε) Αλφαριθμητικός

Σύνθετοι : α) Πίνακας β) Εγγραφή γ) **Λίστα** δ) Σύνολο ε) Σωρός ζ) **Στοίβα** η) **Ουρά** θ) Δένδρο ι) Γράφος.

ΘΕΩΡΙΑ ΚΕΦΑΛΑΙΟ 4

1) Τι ονομάζεται Δομή ακολουθίας; (σελ 46)

2) Τι ονομάζεται Δομή επιλογής if (σελ 46)

3) Σχεδιάστε το Διάγραμμα ροής για την απλή δομή if (σελ 47)

4) Σχεδιάστε το Διάγραμμα ροής για τη δομή if-else (σελ 48)

5) Αναφέρετε τις δομές επιλογής που γνωρίζετε

6) Αναφέρετε τις Δομές επανάληψης που γνωρίζετε

7) Τι τύπος επανάληψης είναι η for και τι τύπος επανάληψης είναι η while (σελ 51)

8) Στην εντολή for χρησιμοποιείται η συνάρτηση _____ για τον καθορισμό των επαναλήψεων. (σελ 51)

9) Η range() είναι μια _____ συνάρτηση της γλώσσας Python, η οποία, ανάμεσα σε άλλα, χρησιμοποιείται για την υπόδειξη του αριθμού των επαναλήψεων που θα εκτελεστούν σε ένα _____. Η δομή της είναι της μορφής range (αρχή, μέχρι, _____) (σελ 52)

10) Σχεδιάστε το Διάγραμμα ροής για τη δομή επανάληψης while (σελ 53)

11) Η εντολή "import random" εισάγει μια βιβλιοθήκη συναρτήσεων για την παραγωγή _____ _____. (σελ 54)

12) Για να ορίσουμε μια δική μας συνάρτηση χρησιμοποιούμε τη χαρακτηριστική λέξη _____, ακολουθεί ένα όνομα που ταυτοποιεί την εκάστοτε συνάρτηση και ένα ζευγάρι παρενθέσεων που μπορούν να περικλείουν ονόματα μεταβλητών, ενώ η γραμμή τελειώνει με _____ _____ (σελ 57)

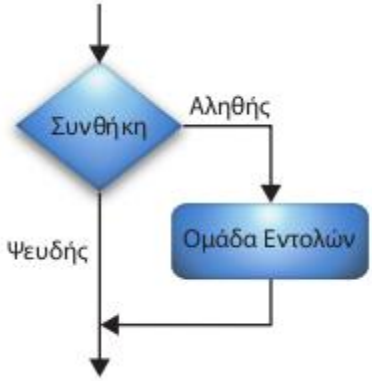
13) Σε μια συνάρτηση οι παράμετροι καθορίζονται μέσα στο ζευγάρι των παρενθέσεων στον ορισμό της συνάρτησης και διαχωρίζονται με _____. Όταν καλούμε τη συνάρτηση, δίνουμε και τις τιμές με τον ίδιο τρόπο, οι οποίες τιμές ονομάζονται _____.

ΑΠΑΝΤΗΣΕΙΣ

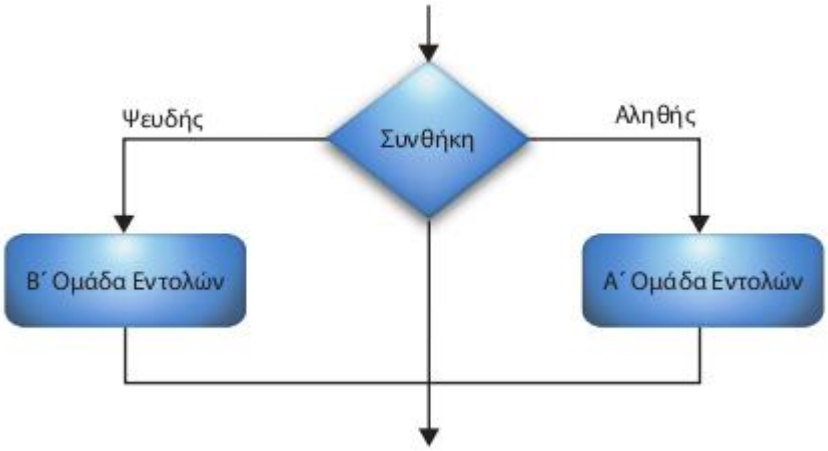
1) Δομή ακολουθίας : Πρόκειται για μια σειρά από εντολές που εκτελούνται η μία μετά την άλλη με τη σειρά. (σελ 46)

2) Η δομή επιλογής if (ΑΝ) χρησιμοποιείται, όταν θέλουμε να εκτελεστεί μια ακολουθία εντολών, μόνον, εφόσον πληρείται μία συγκεκριμένη συνθήκη. (σελ 46)

3) (σελ 47)



4) (σελ 48)



5) α) απλή δομή if, β) δομή if-else, γ) Πολλαπλή Επιλογή (elif), δ) Εμφωλευμένες δομές επιλογής

6) α) for , b) while

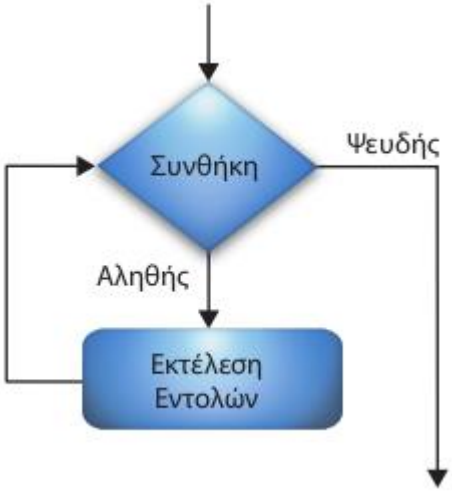
7) a) **for :** προκαθορισμένος, όπου το πλήθος των επαναλήψεων είναι δεδομένο

b) **while :** μη προκαθορισμένος, όπου το πλήθος των επαναλήψεων καθορίζεται κατά τη διάρκεια της εκτέλεσης των εντολών του σώματος της επανάληψης. (σελ 51)

8) α) **range()** (σελ 51)

9) α) ενσωματωμένη, β) βρόχο, γ) βήμα (σελ 52)

10) (σελ 53)



11) α) τυχαίων β) αριθμών. (σελ 54)

12) α) def, β) άνω και κάτω τελεία (:) (σελ 57)

13) α) κόμμα, β) ορίσματα (σελ 58) (διαφοροποιούνται από το κεφάλαιο των συναρτήσεων τα ορίσματα)

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Α1

1) Να γράψετε στο τετράδιό σας τον αριθμό καθεμιάς από τις παρακάτω προτάσεις και δίπλα τη λέξη Σωστό αν είναι σωστή ή τη λέξη Λάθος αν είναι λανθασμένη.

1. Στην Python δεν έχει σημασία αν το όνομα μιας μεταβλητής είναι γραμμένο με κεφαλαία ή μικρά γράμματα.

2. Ένας μετρητής πρέπει να πάρει αρχική τιμή πριν από την εντολή επανάληψης.

3. Μπορούμε, στην ίδια μεταβλητή να εκχωρήσουμε αρχικά μία ακέραια τιμή και μετά μια συμβολοσειρά.

4. Η έκφραση 45 % 10 επιστρέφει 4.5

5. Η εντολή open("words.txt","w") ανοίγει ένα αρχείο για εγγραφή και διατηρεί τα υπάρχοντα περιεχόμενά του.

6. Τα αλφαριθμητικά ή συμβολοσειρές στην Python είναι ακολουθίες από χαρακτήρες που έχουν σταθερό μέγεθος και μη μεταβαλλόμενα περιεχόμενα

7. Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις , τις οποίες τις θεωρεί ως αντικείμενα .

8. Η Λίστα (List) ανήκει στους απλούς Τύπους Δεδομένων .

9. Η συνάρτηση pow (x , y) επιστρέφει το ακέραιο πηλίκο και το ακέραιο υπόλοιπο της πράξης x δια y .

10. Η Συνάρτηση open("words.txt", "a") , αν δεν υπάρχει το αρχείο, το δημιουργεί, ενώ, αν υπάρχει, το ανοίγει σε κατάσταση προσθήκης δεδομένων στο τέλος του.

11. Ο αλγόριθμος της δυαδικής αναζήτησης δεν μπορεί να υλοποιηθεί σε ταξινομημένες λίστες.

12. Έστω ένα αρχείο f το οποίο έχει ανοιχτεί για ανάγνωση. Η εντολή print f.read(1) εκτυπώνει την πρώτη γραμμή του αρχείου στην οθόνη.

13. Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις τις οποίες θεωρεί ως αντικείμενα.

14. Το πλεονέκτημα της απεριόριστης εμβέλειας είναι ότι περιορίζεται η ανεξαρτησία των υποπρογραμμάτων.

15. Στις λίστες τα δεδομένα που υπάρχουν δεν είναι απαραίτητα ίδιου τύπου μεταξύ τους.

16. Οι εντολές που περιλαμβάνονται μέσα στη δομή while θα εκτελεστούν τουλάχιστον μία (1) φορά.

17. Μία συνάρτηση ορίζεται με τη λέξη κλειδί def που την ακολουθεί ένα όνομα το οποίο την ταυτοποιεί, ένα ζεύγος παρενθέσεων, που μπορεί να περιέχει ονόματα μεταβλητών, και τελειώνει με διπλή τελεία (:).

18. Αποδομητής (destructor) ονομάζεται η μέθοδος η οποία καταστρέφει αντικείμενα και ελευθερώνει τη μνήμη.

19. Η μέθοδος fin.tell() επιστρέφει έναν ακέραιο που περιέχει πάντα την τελευταία θέση του αρχείου.

20. Η λίστα, σε αντίθεση με τη συμβολοσειρά, είναι μία δυναμική δομή, στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable).

21. Η δομή της ουράς μπορεί να υλοποιηθεί στην Python με μία λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο.

22. Η λειτουργία της ουράς είναι γνωστή στη βιβλιογραφία ως FIFO (First In First Out).

23. Η αρίθμηση των στοιχείων στις λίστες ξεκινάει από το 1.

24. Η συνάρτηση open ("words.txt" ,"r ") δημιουργεί το αρχείο words.txt αν αυτό δεν υπάρχει.

25. Η δυαδική αναζήτηση εφαρμόζεται στα στοιχεία μιας λίστας τα οποία βρίσκονται σε κάποια λογική διάταξη.

26. Η συνάρτηση random επιστρέφει έναν τυχαίο δεκαδικό ανάμεσα στο 0.0 και στο 1.0 (συμπεριλαμβανομένου του 0.0, αλλά όχι του 1.0)

27. Η δομή της στοίβας μπορεί να υλοποιηθεί στην Python με μια λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο.

28. Η εντολή elif χρησιμοποιείται στην Python για τη σύνταξη σύνθετων δομών επιλογής.

29. Η str() δέχεται οποιαδήποτε αριθμητική τιμή και τη μετατρέπει σε ακέραιο αριθμό.

30. Οι τιμές που μεταβιβάζονται από ένα υποπρόγραμμα σε άλλο, λέγονται παράμετροι.

31. Αν ανοίξουμε ένα υπάρχον αρχείο με τη χρήση του ορίσματος " a" , τυχόν υπάρχοντα περιεχόμενά του θα διαγραφούν.

32. Κάθε συνάρτηση, όταν κληθεί, επιστρέφει πάντα κάποια τιμή.

33. Ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής μπορεί να τροποποιηθεί ώστε να τερματίσει, μόλις διαπιστώσει ότι η λίστα έχει ταξινομηθεί.

34. Οι μεταβλητές που μπορούν να χρησιμοποιηθούν σε οποιοδήποτε τμήμα ενός προγράμματος, ανεξάρτητα από το πού δηλώθηκαν, χαρακτηρίζονται ως καθολικές.

35 . Η δομή while χρησιμοποιείται για μη προκαθορισμένο αριθμό επαναλήψεων .

36 . Η μέθοδος L . pop () προσθέτει ένα στοιχείο στο τέλος της λίστας L .

37. Ο αριθμός 28.2E - 5 είναι ένας αριθμός κινητής υποδιαστολής .

38 . Η int (x) μετατρέπει σε ακέραιο την αριθμητική τιμή x.

39. Ένα αντικείμενο δημιουργείται από μία ειδική μέθοδο που ονομάζεται αποδομητής (de structor) .

40. Στη γλώσσα προγραμματισμού Python για τη χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της .

41 . Στη γλώσσα προγραμματισμού Python χρησιμοποιούμε την εντολή for για να εκτελεστεί ένα τμήμα του κώδικα για έναν καθορισμένο αριθμό επαναλήψεων .

42. Η συνάρτηση pow (4,2) επιστρέφει την τιμή 8.

43 . Η μέθοδος λίστας L.append (object), όπου L το όνομα της λίστας χρησιμοποιείται για προσθήκη του στοιχείου object στο τέλος της λίστας L.

44. Στον αντικειμενοστραφή προγραμματισμό τα χαρακτηριστικά (attributes) ενός αντικειμένου ονομάζονται και μέθοδοι.

45. Το μειονέκτημα των καθολικών μεταβλητών (global) είναι ότι περιορίζουν την ανεξαρτησία των υποπρογραμμάτων.

46. Η κλήση μιας συνάρτησης γίνεται με την εντολή call όνομα_συνάρτησης() .

47. Η εκχώρηση τιμής σε μια μεταβλητή γίνεται με το σύμβολο " == " .

48. Μία συνάρτηση μπορεί να κληθεί και μέσα από μία άλλη συνάρτηση.

49. Η συνάρτηση range (10, 1, - 2) επιστρέφει τη λίστα [10, 8, 6, 4, 2] .

ΑΠΑΝΤΗΣΕΙΣ

1)

1. Σωστό

2. Σωστό

3. Σωστό

4. Λάθος

5. Λάθος

6. Σωστό

7. Σωστό

8. Λάθος

9. Λάθος

10. Σωστό

11. Λάθος

12. Λάθος

13. Σωστό

14. Λάθος

15. Σωστό

16. Λάθος

17. Σωστό

18. Σωστό

19. Λάθος

20. Σωστό

20. Λάθος

22. Σωστό

23. Λάθος

24. Λάθος

25. Σωστό

26. Σωστό

27. Σωστό

28. Σωστό

29. Λάθος

30. Σωστό

31. Λάθος

32. Λάθος

33. Σωστό

34. Σωστό

35. Σωστό

36. Λάθος

37. Σωστό

38. Σωστό

39. Λάθος

40. Σωστό

41. Σωστό

42. Λάθος

43. Σωστό

44. Λάθος

45. Σωστό

46. Λάθος

47. Λάθος

48. Σωστό

49. Σωστό

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Α2

1) Ποιες λίστες αριθμών παράγουν οι παρακάτω συναρτήσεις range;

i. range(5)

ii. range(1, 5)

iii. range(5, -1, -2)

2) Να μετατρέψετε τις παρακάτω προτάσεις σε εντολές στην γλώσσα προγραμματισμού Python.

α. μειώστε την μεταβλητή x κατά 3 μονάδες.

β. εκχωρήστε στην μεταβλητή z το μέσο όρο των μεταβλητών a και b.

γ. αν η μεταβλητή k είναι μεγαλύτερη του μηδενός (0) να εμφανίζει «thetikos arithmos».

δ. να εμφανίσετε τη λέξη «kalimera».

3) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. and	α. Σχεσιακός Τελεστής
2. sqrt()	β. Αριθμητικός Τελεστής
3. or	γ. Πράξη σύζευξης
4. %	δ. Πράξη διάζευξης
5. abs()	ε. Συνάρτηση που επιστρέφει την απόλυτη τιμή
	στ. Συνάρτηση που επιστρέφει την τετραγωνική ρίζα ενός αριθμού

4) Να γράψετε στο τετράδιό σας το αποτέλεσμα που εμφανίζεται στην οθόνη μετά την εκτέλεση του παρακάτω προγράμματος :

Πρόγραμμα	**Οθόνη Η/Υ**
L=[13,5] L=L+[6,24] print L L.append(20) print L L.pop(2) print L L.pop() print L L.insert(0,1) print L	α. ..... β. ..... γ. ..... δ. ..... ε. .....

5) Να χαρακτηρίσετε καθεμιά από τις ακόλουθες λογικές εκφράσεις ως True ή False αν x=5 και y=2:

α . x==6

β . x>4 and y!=1

γ . x<=5 or (y*2<3)

δ . (x>10 and y<3) or (2*y>4)

ε . x>12 and y<10 or (3*y>9)

6) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. float(10)	α. 10.0
2. pow(2,3)	β. 5
3. abs(-10)	γ. 8
4. int(5.6)	δ. 10
	ε. 5.6

7) Να χαρακτηρίσετε τις λογικές εκφράσεις που ακολουθούν, γράφοντας στο τετράδιό σας, δίπλα στο γράμμα που αντιστοιχεί σε κάθε έκφραση, τη λέξη True, αν η πρόταση είναι αληθής, ή τη λέξη False, αν η πρόταση είναι ψευδής.

α. not(4<9)

β. 4<len("καλημέρα")

γ. (4>5) or (9>2)

δ. (4==4) and not(4>9)

ε. pow(3,0)==9-8

8) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. 12 / 4 % 2	α. True
2. not(56<=12)	β. False
3. 45 / 10	γ. 4.5
4. (12 < 11) and(23 > 10)	δ. 1
5. 45.0 / 10	ε. 4
6. 2 * (5 % 4) + 4 / (1 + 3)	στ. 5
	ζ. 3

9) Δίνεται το παρακάτω τμήμα προγράμματοςPython:

```
for i in range (0, 100, 5)
        print i
```

Το τμήμα αυτό του προγράμματος εμφανίζει διαδοχικά τους αριθμούς0, 5, 10, … , 95. Να τροποποιήσετε τον παραπάνω κώδικα έτσι ώστε αυτοί να εμφανίζονται σε αντίστροφη σειρά.

10) Να χαρακτηρίσετε καθεμιά από τις ακόλουθες λογικές εκφράσεις ως True ή False.

α. 34 != 45

β. 56 <= 12

γ. (12<11) and (23>10)

δ. (12<11) or (23>10)

ε. not(56 <= 12)

11) Δίνεται το παρακάτω τμήμα προγράμματος Python:

```
for x in range (A, M, B):
        print x
```

Για καθεμιά από τις παρακάτω περιπτώσεις, να γράψετε στο τετράδιό σας τις τιμές των A, M, B, έτσι ώστε το αντίστοιχο τμήμα προγράμματος να εμφανίζει όλους :

α. τους ακέραιους από 1 μέχρι και 80 (αύξουσα σειρά)

β. τους ακέραιους από 50 μέχρι και 20 (φθίνουσα σειρά)

γ. τους περιττούς ακέραιους από 81 μέχρι και 151 (αύξουσα σειρά)

δ. τους ακέραιους από -50 μέχρι και -5 (αύξουσα σειρά)

ε. τους θετικούς ακέραιους που είναι μικρότεροι του 200 και πολλαπλάσιοι του 7 (αύξουσα σειρά).

12) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1 divmod()	α. Σχεσιακός Τελεστής
2 not	β. Αριθμητικός Τελεστής
3 ==	γ. Τελεστής Λογικής Πράξης
4 %	δ. Συνάρτηση Ενσωματωμένη
	ε. Μη Ενσωματωμένη Συνάρτηση

13) Δίνεται λίστα που περιέχει όλα τα θετικά πολλαπλάσια του 3 μέχρι και το 99. Το τμήμα προγράμματος Python που ακολουθεί αντιγράφει τα στοιχεία της λίστας σε ένα νέο αρχείο κειμένου, με όνομα Pol.txt. Κάθε στοιχείο γράφεται σε μια διαφορετική γραμμή. Στο τμήμα αυτό υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί.

```
pollaplasia=range(3,____(1)____, ____(2)____)
myfile=open("Pol.txt", ____(3)____)
for number in pollaplasia:
    myfile.____(4)____(____(5)____(number) + "\n")
myfile.____(6)____()
```

Να γράψετε στο τετράδιό σας τους αριθμούς(1), (2), (3), (4),(5) και(6) που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό, αυτό που πρέπει να συμπληρωθεί.

14) Σε μια μεταβλητή τύπου ακεραίου(integer) με όνομα x αποθηκεύεται η βαθμολογία ενός μαθητή. Οι επιτρεπτές τιμές είναι από 1 μέχρι και 20. Να γράψετε στο τετράδιό σας ποια από τις παρακάτω εκφράσεις ελέγχει αυτή τη συνθήκη.

α) (x<=1) and (x>=20)

β) (x>=1) or (x<=20)

γ) (x>=1) and (x<=20)

δ) (x<=1) or (x<=20)

15) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. str ()	α. Λογικός τελεστής
2. True	β. Συγκριτικός τελεστής
3. " False "	γ. Λογική τιμή
4. or	δ. Συμβολοσειρά
5. ==	ε. Αριθμητικός τελεστής
	στ . Συνάρτηση μετατροπής μιας τιμής σε συμβολοσειρά

16) Να γράψετε στο τετράδιό σας το αποτέλεσμα που εμφανίζεται στην οθόνη μετά την εκτέλεση καθεμιάς από τις παρακάτω εντολές :

α . range(2,10)

β . range(2,10,3)

γ . a = " abc "

print a*2

δ . x= 2

y=3

print 2 * x+y

ε . a = 2

print a **3

17) Να χαρακτηρίσετε καθεμ ιά από τις ακόλουθες λογικές εκφράσεις ως True ή False αν x =3 και y =1 :

α . not(x>y)

β . (x>5) or (y<2)

γ. (x!=5) and (y!=0)

δ . (x<y) or (x**2>y)

ε . x<len(" abc ")

18) Δίνεται η παρακάτω κλάση:

```
class K inito:
        def __init__(self,  marka,  model):
                self.marka=marka
                self.model=model
        def  fortizi ( self ):
                print  "το κινητό φορτίζει"
```

Με βάση την παραπάνω ορισμένη κλάση:

α) Ποιος είναι ο κατασκευαστής (constructor) της κλάσης .

β) Να προσθέσετε την ιδιότητα cpu _ cores που αντιπροσωπεύει το πλήθος των πυρήνων του επεξεργαστή και την ιδιότητα cam _ resolution που αντιπροσωπεύει την ανάλυση της κάμερας σε Mpixel ώστε να αρχικοποιούνται στον κατασκευαστή .

γ) Να δημιουργήσετε ένα στιγμιότυπο της κλάσης, δηλαδή ένα αντικείμενο με όνομα phone 1 του οποίου οι τιμές των ιδιοτήτων του θα οριστούν κατά τη δημιουργία του ως εξής:

marka = " orange " , model = " S 3" , cpu _ cores = 4 , cam _ resol ution = 10 .

19) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. range(1,10,1)	α. [10]
2. range(10,1,-1)	β. [ ]
3. range(10,-1,-1)	γ. [10,9,8,7,6,5,4,3,2]
4. range(10,0,-1)	δ. [1,2,3,4,5,6,7,8,9]
5. range(10,10,1)	ε. [10,9,8,7,6,5,4,3,2,1,0]
	στ. [10,9,8,7,6,5,4,3,2,1]

20) Να μεταφέρετε και να συμπληρώσετε στο τετράδιό σας τον παρακάτω πίνακα με τα αποτελέσματα των πράξεων μεταξύ τριών μεταβλητών X , Y , Z .

X	**Y**	**Z**	**X>Y and not(Y<=Z)**	**X >2 a nd Y<4 or Z>=5**
10	5	3		
2	5	6		
12	3	5		

21) Ο παρακάτω αλγόριθμος επιστρέφει τη θέση του στοιχείου key αν υπάρχει μέσα στη λίστα array , σε διαφορετική περίπτωση επιστρέφει –1

```
def binarySearch( array, key ) :
    first = 0
    last = len(array) – 1
    pos = (1)
    while first <= last and pos (2) –1 :
        mid = ( first + last ) / 2
        if array[ mid ] == key :
            pos = (3)
        elif array[ mid ] (4) key :
            first = mid + 1
        else :
            last = mid – 1
    return pos
```

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε στο τετράδιό σας τους αριθμούς 1, 2, 3 και 4 που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό αυτό που πρέπει να συμπληρωθεί ώστε να υλοποιείται σωστά η δυαδική αναζήτηση που επιστρέφει τη θέση pos ενός στοιχείου key μέσα σε μία λίστα array .

22) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. x	α. Συμβολοσειρά (string)
2. 256.14	β. Μεταβλητή
3. 28	γ. Μιγαδικός αριθμός
4. True	δ. Αριθμός κινητής υποδιαστολής (float)
5. “True”	ε. Τιμή Λογικού τύπου (boolean)
	στ. Ακέραιος (integer)

23) Δίνονται τα παρακάτω τρία (3) τμήματα προγραμμάτων :

1.	2.	3 .

a = 5
while a - 1!=4:
  print a
  a+=1
	i = 3
	while i<=12:
	  i+=2
	  print i
		y = 2
		while y> - 3:
		  print y
		  y - =1

Να γράψετε τον αριθμό του τμήματος προγράμματος και δίπλα το πλήθος των επαναλήψεων που θα πραγματοποιηθούν σε καθένα από αυτά.

ΑΠΑΝΤΗΣΕΙΣ

1)	**2)**
i. 0, 1, 2, 3, 4 ii. 1, 2, 3, 4 iii. 5, 3, 1	α. x=x-3 β z=(a+b)/2.0 γ. if k>0: print “thetikos arithmos” δ. print “kalimera”

3)	**4)**
1. γ 2. στ 3. δ 4. β 5. ε	α. [13, 5, 6, 24] β. [13, 5, 6, 24, 20] γ. [13, 5, 24, 20] δ. [13, 5, 24] ε. [1, 13, 5, 24]

5)	**6)**
1. False 2. True 3. True 4. False 5. False	1 α 2 γ 3 δ 4 β

7)	**8)**
α. False β. True γ. True δ. True ε. True	1 δ 2 α 3 ε 4 β 5 γ 6 ζ

9)	**10)**
for i in range (95, -5, -5): print i	α. True β. False γ. False δ. True ε. True

11)	**12)**
α. A=1, M=81, B=1 β. A=50, M=19, B=-1 γ. A=81, M=152, B=2 δ. A=-50, M=-6, B=1 ε. A=7, M=200, B=7	1δ 2γ 3α 4β

13)	**14)**
(1) 100 (2) 3 (3) 'w' (4) write (5) str (6) close	γ

15)	**16)**
1.στ 2. γ 3. δ 4. α 5. β	α. 2 3 4 5 6 7 8 9 β. 2 5 8 γ. abcabc δ. 7 ε. 8

17)
α. False β. True γ. True δ. True ε. False

18)
α) Ο κατασκευαστής της κλάσης είναι η ειδική μέθοδος def __init__(self,marka,model) β) def __init__(self,marka,model,cpu_cores,cam_resolution): self.marka=marka self.model=model self.cpu_cores=cpu_cores self.cam_resolution=cam_resolution γ) phone1= Kinito('orange', 'S3', 4 , 10)

19)	**20)**
1. δ 2. γ 3. ε 4. στ 5. β	True False False True False True

21)	**22)**
1) -1 2) == 3) mid 4) <	1. β 2. δ 3. στ 4. ε 5. α

23) 1. καμία **2.** 5 **3.** 5	

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Β

1) Έστω το παρακάτω τμήμα προγράμματος. Να βρείτε τις τιμές όλων των μεταβλητών σε κάθε επανάληψη:

```
K=-5
X=-7
while X<=0:
   M=K+ 4
   if  M<= 0 :
      M,X=X,M
   else:
     X=6*M
   K=K+1
```

2) Έστω το παρακάτω τμήμα προγράμματος. Να βρείτε τις τιμές όλων των μεταβλητών σε κάθε επανάληψη:

```
def F1(Y,X):
   while Y<5:
      print X+2*Y
      Y+=2

X=1
Y=2
F1(X,Y)
```

3) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
def F2(X,Y,Z):

   X=Y-Z
   Z=X+Y
   print X,Y,Z

X=27
Y=2
Z=13

F2(X,Y,Z)
print X,Y,Z

F2(Z, X, Y)

F2(Y, Z, X)
```

4) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
def F3(K, T):
   for x in range(1, K):
      T=T-1
   print T
   return T

S=0
T=0

for  x  in range(5, 1, -2):
   S=S+F3(x, T)

print S
```

5) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
B=[5, 8]

print   B + [10,20,30, range(-5,-8,-2)]
```

6) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
X= [range(5, 20, 5), range(-5,-10,-2), range(3) ]

print X

print X[ 1 ] [ 2 ]

print X[2]
```

7) Έστω το παρακάτω τμήμα προγράμματος. Να βρείτε τις τιμές όλων των μεταβλητών σε κάθε επανάληψη:

```
K=8
L=11
M=0

while L>0:
   if L % 2 ==1:
      M=M+K
   K=K*2
   L=L//2
   print "L",L
   print  "K", K
   print "M", M
print M
```

ΑΠΑΝΤΗΣΕΙΣ

1)

Κ	**Μ**	**Χ**
-5		-7
	-1	
-4	-7	-1
	0	
-3	-1	0
-2	1	6

2)

Χ έξω	**Υ έξω**	**Χ μέσα**	**Υ μέσα**
1	2		
		2	1
		2	3
		2	5

3) -11 2 -9

27 **2** **13**

25 27 52

-14 13 -1

4) -4 -2 -6

5) [5, 8, 10, 20, 30, [-5, -7]]

6) [[5, 10, 15], [-5, -7, -9], [0, 1, 2]]

-9

[0, 1, 2]

7)

L	**Κ**	**Μ**
11	8	0
5	16	8
2	32	24
1	64	24
0	128	88

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Β

1) Δίνεται το παρακάτω πρόγραμμα σε γλώσσα προγραμματισμού Python.

```
i=10
sum=0
while  i<=100  :
        sum =sum + i
        i=i+20
print  i, sum
```

Να γράψετε στο τετράδιό σας :

Α. Ποια είναι η αρχική τιμή της μεταβλητής i;

Β. Ποιες είναι οι διαδοχικές τιμές που θα πάρουν οι μεταβλητές i, sum;

C. Ποιο είναι το περιεχόμενο των μεταβλητών i, sum στο τέλος του προγράμματος;

2) Δίνεται το παρακάτω τμήμα προγράμματος σε γλώσσα προγραμματισμού Python:

```
x=input('Dose timi sto x')
if  x==1  :
        y=x+5
if  x==2  :
        y=x* 5+8
if  x==3  :
        y=2* x-x
if  x==4  :
        y=(x+ x* 5) // 7
if x>4 :
        y=(x // 3)+(x % 3)
print  y
```

Β1. Έστω ότι η θετική ακέραια μεταβλητή εισόδου x (x>0) παίρνει τις ακόλουθες τιμές :
α)1, β)7, γ)4, δ)12, ε)3, στ)2.

Να γράψετε στο τετράδιό σας τις τιμές της μεταβλητής εξόδου y που θα εμφανιστούν στην οθόνη για κάθε μία από τις παραπάνω τιμές εισόδου.

3) Δίνεται ο παρακάτω αλγόριθμος σε λογικό διάγραμμα, όπου οι μεταβλητές Χ, Α, Β, C είναι ακέραιες:

```
X=10
A=20
while  X>0:
        if  A%2 ==0:
                A=A+3
        else:
                A=A-1
        X=X // 2
        print  X, A
B=X+A
C= 2 * A
print  B, C
```

Α) Να μεταφέρετε στο τετράδιό σας και να συμπληρώσετε τον παρακάτω πίνακα με τις τιμές των μεταβλητών Χ, Α που εμφανίζονται σε κάθε επανάληψη.

	X	**A**
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	10	20
1η επανάληψη		
2η επανάληψη		
3η επανάληψη		
4η επανάληψη		

Β) Ποιες είναι οι τιμές των μεταβλητών Β, C που θα εμφανιστούν;

4) Δίνεται το παρακάτω πρόγραμμα σε γλώσσα προγραμματισμού Python:

```
k=32
m=10
while  k>=8 :
        k , a= divmod(k, 2)
        m=m+k
        print k, m
```

Να μεταφέρετε στο τετράδιό σας και να συμπληρώσετε τον παρακάτω πίνακα με τις τιμές των μεταβλητώνk, m, που εμφανίζονται σε κάθε επανάληψη.

	k	**m**
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	32	10
1η επανάληψη		
2η επανάληψη		
3η επανάληψη		

5) Δίνεται το παρακάτω πρόγραμμα σε γλώσσα προγραμματισμού Python:

```
plithos =0
sum=0
x=100.0
z=3
A=[1, 5, 8, 10, 3,19]
while x>5 :
        y=A[plithos]
        plithos=plithos+1
        sum=sum+y
        x, z=divmod(x , 2 )
if sum>27 :
        print 'πλήθος=', plithos
else :
        print 'άθροισμα=', sum
```

Να μεταφέρετε στο τετράδιό σας και να συμπληρώσετε πίνακα με τις τιμές των μεταβλητών plithos, sum, x, z, που παίρνουν σε κάθε επανάληψη.

ΑΠΑΝΤΗΣΕΙΣ

1)

i	sum
10	0
30	10
50	40
70	90
90	160
110	**250**

2)

1 → 6
7 → 3
4 → 3
12 → 4
3 → 3
2 → 18

3)

	X	A	B	C
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	10	20		
1η επανάληψη	5	23		
2η επανάληψη	2	22		
3η επανάληψη	1	25		
4η επανάληψη	0	24	24	28

4)

	k	m
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	32	10
1η επανάληψη	16	26
2η επανάληψη	8	34
3η επανάληψη	4	38

5)

plithos	sum	x	z	y
0	0	100.0	0	
1	1	50	0	1
2	6	25	1	5
3	14	12	0	8
4	24	6	0	10
5	27	3	1	3

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Β

Β1. Δίνεται η παρακάτω συνάρτηση σε γλώσσα προγραμματισμού Python που υλοποιεί το αλγόριθμο της δυαδικής αναζήτησης ενός στοιχείου key μέσα σε μία λίστα array .

```
def    binarySearch( array, key ) :
        first = ( 1 )
        last =  ( 2 )
        found =  ( 3 )
        while first <= last and not found :
                mid = ( first + last ) / 2
                if   array[ mid ] == key :
                        found = True
                elif   array[ mid ] < key :
                        first =  ( 4 )
                else :
                        last =  ( 5 )
        return found
```

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε στο τετράδιό σας τους αριθμούς 1, 2, 3, 4 και 5 που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό αυτό που πρέπει να συμπληρωθεί ώστε να υλοποιείται σωστά η δυαδική αναζήτηση.

Β2. Δίνεται το πρόγραμμα:

```
x=4
S=0
for  i  in  range(10,3,-2):
        S=S+i
        x=S%i
        print "S=",S,
        print " x=", x
```

α. Να γράψετε στο τετράδιό σας ότι ακριβώς εμφανίζεται στην οθόνη κατά την εκτέλεση του παραπάνω προγράμματος.

β. Να ξαναγράψετε το παραπάνω πρόγραμμα, χρησιμοποιώντας την εντολή επανάληψης while αντί της εντολής επανάληψης for έτσι ώστε να εμφανίζει το ίδιο αποτέλεσμα.

Β3. Δίνεται η παρακάτω κλάση:

```
class   ypallilos:
        def __init__(self, eponymo, onoma,misthos):
                self.eponymo=eponymo
                self.onoma=onoma
                self.misthos=misthos
        def afksisemistho(self,amount):
                self.misthos=self.misthos+amount
        def emfanise_stoixeia(self):
                print "Επώνυμο ",self.eponymo
                print "Όνομα ",self.onoma
                print "Μισθός ",self.misthos
```

α . Ποιος είναι ο κατασκευαστής (constructor) της κλάσης;

β Ποιες είναι οι ιδιότητες της κλάσης;

γ. Ποιες είναι οι μέθοδοι της κλάσης.

δ. Να προσθέσετε την ιδιότητα xronia που θα αφορά τα χρόνια υπηρεσίας του υπαλλήλου και να αρχικοποιείται στον κατασκευαστή (Δεν χρειάζεται να εμφανίζεται στην emfanise_stoixeia()) .

ε . Να προσθέσετε μία μέθοδο afxisexronia(self,amount) η οποία να αυξάνει τα χρόνια υπηρεσίας κατά amount.

στ. Να δημιουργήσετε τα παρακάτω στιγμιότυπα της κλάσης:

I. Αντικείμενο με όνομα αντικειμένου yp1 και Επώνυμο “ΑΝΤΩΝΙΟΥ” , όνομα “ΑΝΤΩΝΙΟΣ”, Μισθό 1500 και χρόνια υπηρεσίας 15 .

II. Αντικείμενο με όνομα αντικειμένου yp2 και Επώνυμο “ΠΑΠΑΔΟΠΟΥΛΟΥ” , όνομα “ΕΛΕΝΗ”, Μισθό 2000 και χρόνια υπηρεσίας 20 .

ζ. Να καλέσετε την κατάλληλη μέθοδο, ώστε ο μισθός της ΠΑΠΑΔΟΠΟΥΛΟΥ να αυξηθεί κατά 200.

Β4. Α) Οι παρακάτω συναρτήσεις αφορούν την υλοποίηση της Ουράς. Να τις ξαναγράψετε στο τετράδιό σας συμπληρώνοντας τα κενά:

```
def   enqueue(queue, item) :

        _____________________

def dequeue(queue) :

        _____________________

def isEmpty(queue) :

        _____________________

def createQueue( ) :

        _____________________
```

Β) Τι θα εμφανιστεί στην οθόνη του Η/Υ μετά την εκτέλεση των παρακάτω εντολών

```
A=createQueue()
enqueue(A,10)
enqueue(A,20)
print dequeue(A)
enqueue(A,40)
if not isEmpty(A):
        enqueue(A,87)
        enqueue(A,-6)
print dequeue(A)
print dequeue(A)
```

Γ) Να υλοποιήσετε την δομή δεδομένων “ουρά” ως μια κλάση αξιοποιώντας τις τεχνικές του αντικειμενοστρεφούς προγραμματισμού. Ξαναγράψτε στο τετράδιό σας την παρακάτω κλάση συμπληρώνοντας τα κενά.

```
class Queue :
    def __init__(self ) :
        self.items = [ ]
    def enqueue(self, item) :
        _________________
    def dequeue(self) :
        ____________________
    def isEmpty(self) :
        ____________________
```

Δ) Να ξαναγράψετε το τμήμα προγράμματος του Β χρησιμοποιώντας μόνο την κλάση που δημιουργήσατε στο

```
A=Queue()
A.enqueue(10)
____(1)____
print ____(2)____
____(3)____
if not ____(4)____:
    _____(5)___
    _____(6)___
print ____(7)____
print ____(8)____
```

ΑΠΑΝΤΗΣΕΙΣ

Β1	**Β2 α.**	**Β2 β.**

1. 0
2. len(array) – 1
3. False
4. mid + 1
5. mid – 1
	S= 10 x= 0
	S= 18 x= 2
	S= 24 x= 0
	S= 28 x= 0
		x=4
		S=0
		i=10
		while i>3:
		  S=S+i
		  x=S%i
		  print "S=",S
		  print " x=",x
		  i=i-2

Β3 α.	**Β3 β.**

Ο κατασκευαστής είναι:
def __init__(self, eponymo,onoma,misthos):
  self.eponymo=eponymo
  self.onoma=onoma
  self.misthos=misthos
	Οι ιδιότητες είναι :
	eponymo, onoma, misthos

Β3 γ.	**Β3 ε.**

Οι μέθοδοι είναι:
def afksisemistho(self,amount):
  self.misthos=self.misthos+amount
def emfanise_stoixeia(self):
  print "Επώνυμο ",self.eponymo
  print "Όνομα ",self.onoma
  print "Μισθός ",self.misthos
	def afxisexronia(self,amount):
	  self.xronia=self.xronia+amount

Β3 δ.	**Β3 ζ.**

Έχουμε μόνο αλλαγή στο:
def __init__(self, eponymo, onoma, misthos, **xronia**):
  self.onoma=onoma
  self.eponymo=eponymo
  self.misthos=misthos
  self. xronia = xronia
	yp2. afksisemistho(200)

Β3 στ.	
I. yp1= ypallilos("ΑΝΤΩΝΙΟΥ","ΑΝΤΩΝΙΟΣ",1500,20) II. yp2= ypallilos("ΠΑΠΑΔΟΠΟΥΛΟΥ"," ΕΛΕΝΗ",2000,20)	

Β4 Α

def enqueue(queue, item) :
  queue = queue.append(item)
def dequeue(queue) :
  return queue.pop(0)
def isEmpty(queue) :
  return len(queue) == 0
def createQueue( ) :
  return [ ]

Β4 Β
10 20 40

Β4 Γ

```
class Queue :
      def __init__(self ) :
            self.items = [ ]
     def enqueue(self, item) :
           self.items.append( item )
     def dequeue(self) :
           return self.items.pop( 0 )
    def isEmpty(self) :
          return len(self.items)==0
```

Β4 Δ

```
A=Queue()
A.enqueue(10)
A.enqueue(20)
print A.dequeue()
A.enqueue(40)
if not A.isEmpty():
       A.enqueue(87)
       A.enqueue(-6)
print A.dequeue()
print A.dequeue()
```

ΑΣΚΗΣΕΙΣ ΘΕΩΡΙΑ ΣΥΜΒΟΛΟΣΕΙΡΕΣ ΛΙΣΤΕΣ

1) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: print 3*"test"

Α) Σφάλμα
Β) 3*"test"
Γ) testtesttest
Δ) 3test

2) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
S="new"
print "E" not in S
```

A) True
B) False

3) Τι τιμή θα έχει η σύγκριση: "dimitris">"dimitra"

A) True
B) False

4) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
A="epal"
print len(A)
```

A) 3
B) 4

5) Τα αλφαριθμητικά ή συμβολοσειρές στην Python είναι ακολουθίες από χαρακτήρες που δεν έχουν σταθερό μέγεθος και έχουν μεταβαλλόμενα περιεχόμενα. Σ Λ

6) Μια δομή δεδομένων μπορεί να οριστεί ως ένα σχήμα οργάνωσης σχετικών μεταξύ τους στοιχείων δεδομένων. Σ Λ

7) Τι τιμή θα έχει η σύγκριση: str(480)=='480'

A) True
B) False

8) Τι εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: print int("20")+5

Α) "20"+5
Β) 25
Γ) 205
Δ0 "20"5

9) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
A="TEST ENA"
print A[1],A[6]
```

Α) T E
Β) E N
Γ) T N
Δ) E A

10) Τι εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: print "10"+"10"

Α) 20
Β) "20"
Γ) "10"+"10"
Δ) 1010

11) Ποιο από τα παρακάτω προγράμματα παρουσιάζει τα στοιχεία μιας λίστας L με την αντίστροφη σειρά

```
L=[12,15,2,17,23,48]

N=len(L)
for i in range(1,N-1,-1):
        print L[i]
```

Α)

```
L=[12,15,2,17,23,48]

N=len(L)
for i in range(N,-1,-1):
        print L[i]
```

Β)

```
L=[12,15,2,17,23,48]

N=len(L)
for i in range(N-1,-1,-1):
        print L[i]
```

Γ)

```
L=[12,15,2,17,23,48]

N=len(L)
for i in range(N-1,0,-1):
        print L[i]
```

Δ)

12) Η συνάρτηση list (String) επιστρέφει μια λίστα με στοιχεία τους χαρακτήρες της συμβολοσειράς string.

A) True
B) False

13) Έστω ότι έχουμε τη λίστα L=[2,5,8]. Ποιο θα είναι το αποτέλεσμα της σύγκρισης: 8 in L

A) True
B) False

14) Αν έχουμε μία λίστα L, η εντολή L+=[10] προσθέτει το στοιχείο 10 στην αρχή της λίστας L Σ Λ

15) Η αρίθμηση των στοιχείων σε μια λίστα ξεκινάει από το 1. Σ Λ

16) Μπορούμε να έχουμε σε μια λίστα ακόμα και στοιχεία διαφορετικού τύπου π.χ. αριθμούς, συμβολοσειρές κλπ. Σ Λ

17) Θεωρούμε ότι έχουμε μία λίστα L. Η εντολή L.append(object)

Α) Αφαιρεί το τελευταίο στοιχείο της λίστας L
Β) Προσθέτει το στοιχείο object στο τέλος της λίστας L.
Γ) Προσθέτει το στοιχείο object στην αρχή της λίστας L.
Δ) Όλα τα παραπάνω

18) Δίνεται το παρακάτω τμήμα προγράμματος. Τι θα εμφανιστεί στην οθόνη του υπολογιστή;

```
L=[1,5,7,9,2]
L.append(13)
L.pop(2)
L.pop()
L.insert(2,4)
print L
```

Α) [1, 5, 4, 9, 2]
Β) [1, 7, 4, 9, 2]
Γ) [1, 5, 9, 2, 2]
Δ) [1, 7, 9, 2, 2]

19) Δίνεται η λίστα L=[4,5,9] και η λίστα K=[1,2,3]. Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: print L+K

Α) [1,2,3,4,5,6]
Β) [4,5,9][1,2,3]
Γ) [4,5,9]+[1,2,3]
Δ) [4,5,9,1,2,3]

20) Η λίστα είναι μια δυναμική δομή στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable) Σ Λ

21) Ποια από τις παρακάτω συναρτήσεις δέχεται μία λίστα σαν είσοδο και επιστρέφει σωστά τη μέγιστη τιμή της;

```
def MAX(L):
        N=len(L)
        Max=L[0]
        for i in range(N):
                if Max>L[i]:
                        Max=L[i]
        return Max
```

Α)

```
def MAX(L):
        N=len(L)
        Max=L[0]
        for i in range(N):
                if Max<L[i]:
                        Max=L[i]
        return Max
```

Β)

```
def MAX(L):
        N=len(L)
        Max=L[0]
        for i in range(N-1):
                if Max>L[i]:
                        Max=L[i]
        return Max
```

Γ)

```
def MAX(L):
        N=len(L)
        Max=L[0]
        for i in range(N-1):
                if Max<L[i]:
                        Max=L[i]
        return Max
```

Δ)

22) Δίνεται το παρακάτω τμήμα προγράμματος. Τι θα εμφανιστεί στην οθόνη του υπολογιστή ;

```
def enqueue(queue, item) :
        queue = queue.append( item )

def dequeue(queue) :
        return queue.pop( 0 )

def isEmpty(queue) :
        return len(queue) == 0


def createQueue( ) :
        return [ ]

L=createQueue()
enqueue(L,3)
enqueue(L,5)
dequeue(L)
enqueue(L,8)
enqueue(L,11)
dequeue(L)

print L
```

Α) [11, 8]
Β) [3, 8]
Γ) [8, 3]
Δ) [8, 11]

23) Δίνεται το παρακάτω τμήμα προγράμματος. Τι θα εμφανιστεί στην οθόνη του υπολογιστή ;

```
def push(stack, item) :
        stack.append( item )

def pop(stack) :
        return stack.pop( )

def isEmpty(stack) :
        return len(stack) == 0

def createStack( ) :
        return [ ]

L=createStack()
push(L,5)
push(L,10)
push(L,13)
pop(L)
push(L,15)
print L
```

Α) [10, 13, 15]
Β) [5, 10, 15]
Γ) [15, 10, 5]
Δ) [15, 13, 10]

24) Η δομή της Στοίβας μπορεί να υλοποιηθεί στην Python με μια λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο Σ Λ

25) Το σύνολο των επικεφαλίδων των συναρτήσεων το οποίο είναι διαθέσιμο στον προγραμματιστή που χρησιμοποιεί τη Στοίβα, το ονομάζουμε διεπαφή ή διασύνδεση (interface) της δομής αυτής Σ Λ

26) Στην δομή δεδομένων Ουρά οι λειτουργίες εισαγωγής και εξαγωγής είναι γνωστές στη βιβλιογραφία ως ώθηση (push) και απώθηση (pop). Σ Λ

27) Η λειτουργία της Ουράς χαρακτηρίζεται ως LIFO (Last In First Out) Σ Λ

28) Μια δομή δεδομένων που χρησιμοποιείται για την μοντελοποίηση και προσομοίωση πραγματικών φαινομένων, είναι η δομή της Ουράς.Τα φαινόμενα αυτά μελετώνται από διάφορους κλάδους των Μαθηματικών και της Πληροφορικής, όπως είναι η Θεωρία Ουρών (Queueing theory) και η Επιχειρησιακή 'Ερευνα (Operations Research) Σ Λ

29) Η Στοίβα είναι μία δομή δεδομένων στην οποία το στοιχείο που προστέθηκε τελευταίο είναι και το πρώτο που θα εξαχθεί Σ Λ

ΑΠΑΝΤΗΣΕΙΣ

1	testtesttest
2	True
3	True
4	4
5	ΛΑΘΟΣ
6	ΣΩΣΤΟ
7	True
8	25
9	E N
10	1010
11	Γ
12	True
13	True
14	ΛΑΘΟΣ
15	ΛΑΘΟΣ
16	ΣΩΣΤΟ
17	Β
18	Α
19	Δ
20	ΣΩΣΤΟ
21	Β
22	Δ
23	Β
24	ΣΩΣΤΟ
25	ΣΩΣΤΟ
26	ΛΑΘΟΣ
27	ΛΑΘΟΣ
28	ΣΩΣΤΟ
29	ΣΩΣΤΟ

ΘΕΩΡΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

1) Σύμφωνα με τον Τμηματικό Προγραμματισμό , μπορούμε να γράψουμε ένα πρόγραμμα ως ένα σύνολο από μικρότερα κομμάτια προγράμματος Σ Λ

2) Μια συνάρτηση μπορεί να καλείται από διάφορα σημεία του κύριου προγράμματος απαγορεύεται όμως να καλείται μέσα από μια άλλη συνάρτηση Σ Λ

3) Ποια είναι τα βασικά χαρακτηριστικά που πρέπει να έχει ένα υποπρόγραμμα;

Α) Να έχει μόνο ένα σημείο εισόδου από το οποίο δέχεται τα δεδομένα του.

Β) Το (υπο)πρόγραμμα το οποίο καλεί ένα άλλο υποπρόγραμμα σταματάει την εκτέλεσή του όσο εκτελείται το καλούμενο υποπρόγραμμα. Μόνο ένα υποπρόγραμμα μπορεί να εκτελείται σε μια χρονική στιγμή.

Γ) Ο έλεγχος επιστρέφει στο (υπο)πρόγραμμα το οποίο καλεί, όταν το καλούμενο υποπρόγραμμα σταματήσει να εκτελείται.

Δ) Όλα τα παραπάνω

4) Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις, τις οποίες τις θεωρεί ως αντικείμενα. Σ Λ

5) Η εμβέλεια (scope) μιας μεταβλητής αναφέρεται στο τμήμα του προγράμματος που μπορεί αυτή να έχει πρόσβαση Σ Λ

6) Οι μεταβλητές που ισχύουν μόνο για το υποπρόγραμμα στο οποίο δηλώθηκαν ονομάζονται καθολικές (global) Σ Λ

7) Ένα άρθρωμα είναι ένα Python αντικείμενο. Πιο απλά είναι ένα αρχείο αποτελούμενο από κώδικα Python και μπορεί να ορίσει συναρτήσεις, κλάσεις και μεταβλητές. Σ Λ

8) Για να χρησιμοποιήσουμε ένα άρθρωμα στο πρόγραμμά μας, θα πρέπει να το "εισάγουμε" σε αυτό. Αυτό γίνεται με τη δήλωση import. Σ Λ

9) from math import sqrt

Μετά την παραπάνω εντολή, η εντολή που ακολουθεί είναι σωστή;

print sqrt(25)

Σ Λ

10) Ένα άρθρωμα έχει μέσα του πολλές πρότυπες βιβλιοθήκες Σ Λ

11)

```
def allaxe(x):
        x=13

x=2
allaxe(x)
print x
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

Α) 13

Β) 2

Γ) Τίποτα

Δ) Θα εμφανίζει συντακτικό λάθος

12)

```
def allaxe(x):
        x=10
        print x,y

x=2
y=3

allaxe(x)
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

Α) Τίποτα

Β) Θα εμφανίζει συντακτικό λάθος

Γ) 2 3

Δ) 10 3

13)

```
def allaxe(x):
        x=4
        y=5
        print "Από μέσα ",x,y

x=10
y=3

allaxe(x)
print "Από έξω ",x,y
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

Α) Από μέσα 4 5
Από έξω 10 3

Β) Από μέσα 10 3
Από έξω 10 3

Γ) Από μέσα 10 5
Από έξω 10 3

Δ) Από μέσα 4 3
Από έξω 10 3

14)

```
def allaxe(x):
        global y
        x=2
        y=200
        print "Από μέσα ",x,y

x=1
y=7

allaxe(x)
print "Από έξω ",x,y
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

Α) Από μέσα 2 200
Από έξω 1 7

Β) Από μέσα 2 200

Από έξω 2 200

Γ) Από μέσα 2 200

Από έξω 1 200

Δ) Από μέσα 2 7

Από έξω 1 7

15)

```
import random

for i in range(50):
        print random.randrange(6)
```

Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος;

Α) Θα εμφανίζει συντακτικό λάθος

Β) Θα εμφανίσει 50 τυχαίους ακέραιους από το 0 έως και το 6

Γ) Θα εμφανίσει 50 τυχαίους ακέραιους από το 1 έως και το 6

Δ) Θα εμφανίσει 50 τυχαίους ακέραιους από το 0 έως και το 5

ΑΠΑΝΤΗΣΕΙΣ

1	Σ
2	Λ
3	Όλα τα παραπάνω
4	Σ
5	Σ
6	Λ
7	Σ
8	Σ
9	Σ
10	Λ
11	2
12	10 3
13	Από μέσα 4 5 Από έξω 10 3
14	Από μέσα 2 200 Από έξω 1 200
15	Θα εμφανίσει 50 τυχαίους ακέραιους από το 0 έως και το 5

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Γ ΚΑΙ Δ

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2009

1) Μία εταιρεία κινητής τηλεφωνίας ακολουθεί ανά μήνα την πολιτική τιμών, που φαίνεται στον παρακάτω πίνακα:

Πάγιο 4,5 €	
ΑΡΙΘΜΟΣ ΜΗΝΥΜΑΤΩΝ	**ΧΡΕΩΣΗ ΑΝΑ ΜΗΝΥΜΑ**
1-50	0,10 €
51-150	0,08 €
από 151 και άνω	0,05 €

Να αναπτύξετε αλγόριθμο σε Python ο οποίος:

α. Να διαβάζει τον αριθμό των μηνυμάτων ενός συνδρομητή στο τέλος ενός μήνα.

β. Να υπολογίζει τη μηνιαία χρέωση του συνδρομητή. Η χρέωση των μηνυμάτων είναι κλιμακωτή με βάση τον παραπάνω πίνακα.

γ. Να εμφανίζει (τυπώνει) τη λέξη «ΧΡΕΩΣΗ» και τη μηνιαία χρέωση του συνδρομητή.

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2010

2) Να γράψετε στο τετράδιό σας πρόγραμμα σε γλώσσα Python, το οποίο:

Δ1. Να διαβάζει το επώνυμο του υποψηφίου με τη χρήση κατάλληλου μηνύματος.

Δ 2. Να διαβάζει τους βαθμούς στα τέσσερα μαθήματα του υποψηφίου με τη χρήση κατάλληλου μηνύματος.

Δ 3. Να υπολογίζει το μέσο όρο και να κατατάσσει τον υποψήφιο σε μια από τις τέσσερις κατηγορίες, ανάλογα με την επίδοσή του, δηλαδή:

κλίμακα	1-150	151-300	301-350	351-400
χαρακτηρισμός	D	C	B	A

Δ 4. Να εμφανίζει σε ποια κατηγορία ανήκει ο κάθε υποψήφιος.

Δ5. Η επανάληψη θα συνεχίζεται έως ότου στη θέση του επωνύμου γραφτεί η λέξη: τέλος .

Υποδείξεις :

α. Η αποδεκτή βαθμολογία σε κάθε μάθημα είναι από 1 ... 400 και δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών

β. Όπου απαιτείται επανάληψη, να γίνει χρήση μόνο της εντολής while.

ΑΣΚΗΣΗ 1

```
M=input("Δώσε αριθμό μηνυμάτων ")

if M>=1 and M<= 50:
   Xr=M*0.10 + 4.5

if M>=51 and M<= 150:
   Xr=50*0.10 + (M-50)*0.08 + 4.5

if M>=151:
   Xr=50*0.10 + 100*0.08 + (M-150)*0.05 + 4.5

print "ΧΡΕΩΣΗ ",Xr
```

ΑΣΚΗΣΗ 2

```
On=raw_input("Δώσε ονομα")

while On!="τελος":

   V1=input("Δώσε βαθμό 1")

   V2=input("Δώσε βαθμό 2")

   V3=input("Δώσε βαθμό 3")

   V4=input("Δώσε βαθμό 4")

   MO=(V1+V2+V3+V4)/4.0

   if MO>=1 and MO<=150:
      print "D"
   if MO>=151 and MO<=300:
      print "C"
   if MO>=301 and MO<=350:
      print "B"
   if MO>=351 and MO<=400:
      print "A"

   On=raw_input("Δώσε ονομα")
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2011

1) Το σύνολο των σχολείων μιας πόλης αποφάσισε να επισκεφθούν το Ενυδρείο της περιοχής τους. Η χρέωση για την είσοδο των μαθητών ανά σχολείο θα γίνει σύμφωνα με τον παρακάτω πίνακα:

Τιμή εισιτηρίου ανά μαθητή 10 Ευρώ	
1 έως και 20 μαθητές	Κανένα δωρεάν εισιτήριο
21 έως και 40 μαθητές	Δώρο το κόστος 5 εισιτηρίων
41 και άνω μαθητές	Δώρο το κόστος 9 εισιτηρίων

Να γραφεί αλγόριθμος ο οποίος:

Γ1. Να διαβάζει το πλήθος Ν των σχολείων της πόλης.

Γ 2. Για καθένα από τα σχολεία να διαβάζει το όνομά του και το πλήθος των μαθητών που θα συμμετάσχουν.

Γ 3. Να υπολογίζει το πληρωτέο ποσό κάθε σχολείου ανάλογα με το πλήθος των μαθητών του.

Γ4. Να εμφανίζει το όνομα και το ποσό πληρωμής κάθε σχολείου.

Υποδείξεις :

α. Ο υπολογισμός δεν γίνεται κλιμακωτά. Για παράδειγμα σχολείο 50 μαθητών θα τύχει δώρου 9 εισιτηρίων και θα πληρώσει 410 Ευρώ.

β. Δεν χρειάζεται να γίνει έλεγχος ορθότητας τιμών.

2) Μια εταιρεία παραγωγής γραφικής ύλης που διανέμει τα προϊόντα της μέσω πωλητών της, επιθυμεί στο τέλος της χρονιάς να ελέγξει την απόδοσή τους.

Να γράψετε στο τετράδιό σας πρόγραμμα σε γλώσσα Python το οποίο:

Δ1. Να διαβάζει το όνομα του πωλητή.

Δ2. Το ανωτέρω (Δ 1) να επαναλαμβάνεται έως ότου δοθεί για όνομα πωλητή η τιμή 'ΤΕΛΟΣ'.

Δ3. Κατά τη διάρκεια της επανάληψης να διαβάζεται το ποσό των ετήσιων πωλήσεων κάθε πωλητή και μετά το τέλος των επαναλήψεων να έχουν υπολογιστεί τα ακόλουθα:

α) Το πλήθος των πωλητών με πωλήσεις >=50000 Ευρώ.

β) Το πλήθος των πωλητών με πωλήσεις <50000 Ευρώ.

γ) Το συνολικό ποσό των πωλήσεων όλων των πωλητών.

Δ4. Στο τέλος των επαναλήψεων να:

α) τυπώσει το πλήθος των πωλητών με πωλήσεις >=50000 Ευρώ

β) τυπώσει το πλήθος των πωλητών με πωλήσεις <50000 Ευρώ

γ) τυπώσει το συνολικό ποσό των πωλήσεων όλων των πωλητών

δ) υπολογίσει και να τυπώσει το μέσο όρο των πωλήσεων όλων των πωλητών.

Υποδείξεις :

α. Η εταιρεία διαθέτει τουλάχιστον έναν πωλητή.

β. Για την επανάληψη να γίνει χρήση της εντολής while.

γ. Στις εντολές εισόδου και εξόδου να υπάρχουν τα κατάλληλα μηνύματα.

ΑΣΚΗΣΗ 1

```
N=input("Δώσε αριθμό σχολείων")

for x in range (N):
   on=raw_input(" Δώσε όνομα σχολείου")
   a=input(" Δώσε αριθμό μαθητών")

   if a >=1 and a<=20:
      Xr=a*10

   if a >=21 and a<=40:
      Xr=(a-5)*10

   if a >=41 :
      Xr=(a-9)*10

   print "Το σχολείο ",on," θα πληρώσει ",Xr
```

ΑΣΚΗΣΗ 2

```
c1=0
c2=0
S=0.0

ON=input("Δώσε όνομα πωλητή")

while ON!="ΤΕΛΟΣ":

   E=input("Δώσε ετήσιες πωλήσεις ")

   if E>=50000:
      c1+=1
   else:
      c2+=1

   S=S+E

   ON=input("Δώσε όνομα πωλητή")

print "πλήθος πωλητών με πωλήσεις >=50000 ",c1

print "πλήθος πωλητών με πωλήσεις <50000 ",c2

print "συνολικό ποσό πωλήσεων ",S

print "μέσος όρος πωλήσεων ",S/(c1+c2)
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2012

1) Μια επιχείρηση έχει 50 υπαλλήλους. Η διοίκηση έκλεισε συμφωνία για την παροχή πακέτων σύνδεσης με πρόσβαση στο Διαδίκτυο μέσω κινητού τηλεφώνου για κάθε υπάλληλο. Το πακέτο έχει **πάγιο 3 ευρώ** και η χρέωση είναι κλιμακωτή με βάση τον παρακάτω πίνακα:

Όγκος δεδομένων σε GB (Gigabyte)	**Ευρώ ανά GB**
έως και 2	10
3 έως και 6	2,5
7 και άνω	1,5

Να γραφεί αλγόριθμος σε Python, ο οποίος:

Γ1. Να διαβάζει το όνομα και τον όγκο δεδομένων σε GB κάθε υπαλλήλου.

Γ 2. Να υπολογίζει για καθέναν τη χρέωσή του, με το πάγιο.

Γ 3. Να εμφανίζει το όνομα και τη συνολική του χρέωση.

Γ 4. Να υπολογίζει και να εμφανίζει το μέσο όρο της χρέωσης όλων των υπαλλήλων, με το πάγιο.

Υποδείξεις :

α. Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν τα κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

β. Ο όγκος δεδομένων παίρνει ακέραιες τιμές.

γ. Παράδειγμα χρέωσης: ένας υπάλληλος με όγκο δεδομένων 5GB θα χρεωθεί 27.5 ευρώ χωρίς το πάγιο.

2) Μια ναυτιλιακή εταιρεία σε ένα οχηματαγωγό της πλοίο και μόνο σε σχέση με τα οχήματα, εφαρμόζει την τιμολογιακή πολιτική που φαίνεται στον παρακάτω πίνακα:

Τύπος οχήματος	**Χρέωση ανά όχημα**
Μηχανή	10 ευρώ
Αυτοκίνητο ΙΧ	20 ευρώ
Φορτηγό	30 ευρώ

Ο οδηγός δεν πληρώνει εισιτήριο, ενώ κάθε επιπλέον επιβάτης του οχήματος πληρώνει 5 ευρώ.

Να γραφεί πρόγραμμα σε Python, το οποίο:

Δ1. Να διαβάζει τον τύπο του οχήματος (‘Μ ’ για μηχανή, ‘Α ’ για αυτοκίνητο, ‘Φ’ για φορτηγό) και τον αριθμό των επιβατών του (μαζί με τον οδηγό).

Δ2. Να υπολογίζει το κόστος για κάθε όχημα, στο οποίο να συμπεριλαμβάνεται και το κόστος των επιβατών.

Δ3. Η διαδικασία (Δ1-Δ2) επαναλαμβάνεται για όλα τα οχήματα και μέχρι να δοθεί η τιμή ‘ΤΕΛΟΣ’ στον τύπο του οχήματος.

Δ4. Μετά την επανάληψη να εμφανίζονται:

α) Το πλήθος των φορτηγών.

β) Η συνολική χρέωση όλων των οχημάτων μαζί με τους επιβάτες τους.

Υποδείξεις :

α. Να γίνει έλεγχος ορθότητας τιμών στον τύπο του οχήματος.

β. Το πρόγραμμα δεν ασχολείται με τους επιβάτες άνευ οχήματος.

γ. Για την επανάληψη να γίνει χρήση της εντολής while.

δ. Στο πλοίο εισέρχεται τουλάχιστον ένα όχημα, ενώ όλα τα οχήματα είναι με οδηγό.

ΑΣΚΗΣΗ 1

```
S=0.0
for x in range(50):
   ON=raw_input("Δώσε όνομα")
   D=input("Δώσε όγκο δεδομένων  ")

   if D>=1 and D<= 2:
      Xr=D*10 + 3

   if D>=3 and D<= 6:
      Xr=2*10 + (D-2)*2.50 + 3

   if D>=7:
      Xr=2*10 + 5*2.50 + (D-7)*1.50 + 3

   print "Ο Υπάλληλος ",ON, "θα πληρώσει ",Xr

   S=S+Xr

print "μέσος όρος χρέωσης όλων των υπαλλήλων",Xr/50
```

ΑΣΚΗΣΗ 2

```
S=0
SF=0

OX=raw_input("Δώσε τύπο οχήματος")
while OX not in ["Μ", "Α","Φ"]:
   OX=raw_input("Δώσε ΣΩΣΤΟ τύπο οχήματος")

while OX!="ΤΕΛΟΣ" :

   A=input("Δώσε αριθμό επιβατών ")

   if OX=="Μ":
      K=10 + 5*(A-1)

   if OX=="Α":
      K=20 + 5*(A-1)

   if OX=="Φ":
      K=30 + 5*(A-1)
      SF+=1

   S=S+K

   OX=raw_input("Δώσε τύπο οχήματος")
   while OX not in ["Μ", "Α","Φ"]:
      OX=raw_input("Δώσε ΣΩΣΤΟ τύπο οχήματος")

print "συνολική χρέωση όλων των οχημάτων",S

print "πλήθος των φορτηγών:",SF
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2013

1) Ένα σχολείο πρόκειται να πάει μια εκπαιδευτική επίσκεψη. Στην προσφορά, που έγινε από ένα ταξιδιωτικό γραφείο, περιλαμβάνεται το κόστος ανά μαθητή, χωρίς τη διατροφή ξενοδοχείο, όπως φαίνεται στον παρακάτω πίνακα:

Αριθμός μαθητών	**Κόστος ανά μαθητή**
Από 1 έως και 100	80€
Από 101 και πάνω	60€

Αν το σχολείο επιλέξει, να έχουν οι μαθητές και διατροφή στο ξενοδοχείο, το κόστος αυξάνεται κατά 30€ ανά μαθητή.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Να διαβάζει τον αριθμό των μαθητών, που θα συμμετάσχουν στην εκπαιδευτική επίσκεψη. Επίσης, να διαβάζει την απάντηση του σχολείου σχετικά με τη διατροφή των μαθητών (‘ΝΑΙ’ αν επιθυμούν, ‘ΟΧΙ’ αν δεν επιθυμούν διατροφή).

Γ2. Να υπολογίζει και να εμφανίζει το συνολικό κόστος της εκπαιδευτικής επίσκεψης, χωρίς τη διατροφή.

Γ3. Να υπολογίζει το συνολικό κόστος της διατροφής, στην περίπτωση που το σχολείο την έχει επιλέξει.

Γ4. Να υπολογίζει και να εμφανίζει το συνολικό κόστος της εκπαιδευτικής επίσκεψης.

Υποδείξεις :

α. Να γίνει έλεγχος ορθότητας τιμών στην απάντηση του σχολείου σχετικά με τη διατροφή

β. Αν ο αριθμός των μαθητών είναι μικρότερος ή ίσος του 100, τότε όλοι οι μαθητές θα πληρώσουν από 80€ ο καθένας, ενώ αν ο αριθμός τους είναι μεγαλύτερος του 100, τότε όλοι οι μαθητές θα πληρώσουν από 60€ ο καθένας.

2) Μια αεροπορική εταιρεία πρόκειται να δρομολογήσει μία πτήση με αεροπλάνο 100 θέσεων. Τα λειτουργικά έξοδα της πτήσης είναι 5.000€ και όλα τα εισιτήρια έχουν διατεθεί. Η τιμή του εισιτηρίου, που πλήρωσε ο κάθε επιβάτης, μπορεί να κυμαινόταν από 20€ έως και 200€.

Να γράψετε στο τετράδιό σας, έναν αλγόριθμο σε Python, ο οποίος:

Δ1. Να διαβάζει το όνομα του κάθε επιβάτη.

Δ2. Να διαβάζει την τιμή του εισιτηρίου του κάθε επιβάτη με έλεγχο ορθότητας τιμών.

Δ3. Να υπολογίζει τις συνολικές εισπράξεις της εταιρείας από τη συγκεκριμένη πτήση.

Δ4. Να υπολογίζει και να εμφανίζει το όνομα του επιβάτη με τη μικρότερη τιμή εισιτηρίου. Να θεωρήσετε, ότι η μικρότερη τιμή είναι μοναδική.

Δ5. Να εμφανίζει το μήνυμα «ΚΕΡΔΟΣ», αν οι συνολικές εισπράξεις είναι μεγαλύτερες από τα λειτουργικά έξοδα ή το μήνυμα «ΖΗΜΙΑ», αν οι συνολικές εισπράξεις είναι μικρότερες από τα λειτουργικά έξοδα ή το μήνυμα «ΜΗΔΕΝΙΚΟ ΑΠΟΤΕΛΕΣΜΑ», αν είναι ίσα.

Υποδείξεις :

Δε χρειάζεται να υπάρχουν τα κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου, εκτός του ερωτήματος Δ 5.

ΑΣΚΗΣΗ 1

```
N=input("Δώσε αριθμό μαθητών")

D=raw_input("Θα υπάρχει διατροφή των μαθητών;")
while D not in ["ΝΑΙ","ΟΧΙ"]:
   D=raw_input("Δώσε ΝΑΙ ή ΟΧΙ")


if N>=1 and N<=100:
   Xr=N*80
if N>100:
   Xr=N*60

print "κόστος χωρίς τη διατροφή",Xr

if D=="ΝΑΙ":
   XD=30*N
   print "κόστος διατροφή",XD

print "συνολικό κόστος της εκπαιδευτικής επίσκεψης",Xr+XD
```

ΑΣΚΗΣΗ 2

```
min1=201

S=0
for x in range(100):
   ON=raw_input("Δώσε ονομα")
   T=input("Δώσε τιμή εισιτηρίου ")
   while T<20 or T>200:
      T=input("Δώσε ΣΩΣΤΗ τιμή εισιτηρίου ")

   S=S+T

   if T<min1:
      min1=T
      onoma=ON

print "Ο επιβάτης",onoma, "έχει το φθηνότερο εισιτήριο"

if S<5000:
   print "ΚΕΡΔΟΣ"
if S>5000:
   print "ΖΗΜΙΑ"
if S==5000:
   print "ΜΗΔΕΝΙΚΟ ΑΠΟΤΕΛΕΣΜΑ"
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2014

1) Μια ασφαλιστική εταιρία στον κλάδο του αυτοκινήτου προσφέρει τις παρακάτω τιμές ετήσιου ασφαλίστρου ανάλογα με τον κυβισμό του αυτοκινήτου.

Κυβισμός σε cc	**Τιμή ετήσιου ασφαλίστρου σε €**
Έως και 1000	150
Από 1001 έως και 2000	200
Πάνω από 2000	300

Σε περίπτωση που ο οδηγός είναι νέος, δηλαδή έχει ηλικία μικρότερη ή ίση των 23 ετών, τότε τα ασφάλιστρα αυξάνονται κατά 40€. Στην τιμή που προκύπτει από όλα τα προηγούμενα προστίθεται ΦΠΑ 23%.

Να γραφεί αλγόριθμος σε Python, ο οποίος:

Γ1. Να διαβάζει τον κυβισμό ενός αυτοκινήτου καθώς και την ηλικία του οδηγού.

Γ2. Να υπολογίζει τα ετήσια ασφάλιστρα του αυτοκινήτου με βάση τον κυβισμό του και την ηλικία του οδηγού.

Γ3. Να υπολογίζει και να εμφανίζει το τελικό ποσό ετήσιων ασφαλίστρων συμπεριλαμβανομένου του ΦΠΑ.

Υποδείξεις :

Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

2) Για τη διεξαγωγή των Πανελλαδικών Εξετάσεων των ΕΠΑ.Λ. σήμερα, το Υπουργείο Παιδείας & Θρησκευμάτων έχει ορίσει 143 εξεταστικά κέντρα σε όλη τη χώρα.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο:

Δ1. Θα διαβάζει για καθένα εξεταστικό κέντρο:

α. το νομό στον οποίο βρίσκεται.

β. το πλήθος των μαθητών που έχουν δικαίωμα να προσέλθουν στις εξετάσεις.

γ. το πλήθος των απόντων μαθητών.

Δ3. Να εμφανίζει το νομό που βρίσκεται το μεγαλύτερο εξεταστικό κέντρο (δηλαδή αυτό με το μεγαλύτερο πλήθος μαθητών, οι οποίοι έχουν δικαίωμα να προσέλθουν στις εξετάσεις). Να θεωρήσετε ότι αυτό το εξεταστικό κέντρο είναι μοναδικό.

Δ4. Να εμφανίζει το συνολικό αριθμό των μαθητών που προσήλθαν στις εξετάσεις σε όλα τα εξεταστικά κέντρα της χώρας.

Δ5. Να εμφανίζει το πλήθος των εξεταστικών κέντρων στα οποία προσήλθαν για τις εξετάσεις όλοι οι μαθητές, δηλαδή δεν έχουν κανένα απόντα.

Υπόδειξη :

Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν τα κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

ΑΣΚΗΣΗ 1

```
K=input("Δώσε κυβισμό αυτοκινήτου ")

H=input("Δώσε την ηλικία του οδηγού")

if K>=1 and K<=1000:
   A=150

if K>=1001 and K<=2000:
   A=200

if K>2000:
   A=300

if H<=23:
   A=A+40

SA=A + A*23/100

print "τελικό ποσό ετήσιων ασφαλίστρων ",SA
```

ΑΣΚΗΣΗ 2

```
S=0
c=0

for x in range(143):
   N=raw_input("Δώσε νομό")
   M=input("Δώσε πλήθος μαθητών ")
   A=input("Δώσε πλήθος των απόντων μαθητών")

   S=S+(M-A)

   if A==0:
      c=c+1

print "Σύνολο μαθητών που προσήλθαν στις εξετάσεις",S

print "πλήθος εξεταστικών κέντρων χωρις απόντες",c
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2015

1) Ένα κατάστημα ηλεκτρονικών ειδών προσφέρει τα παρακάτω ποσοστά εκπτώσεων, ανάλογα με τον τρόπο πληρωμής που θα επιλέξει ο πελάτης:

Τρόπος πληρωμής	Ποσοστό έκπτωσης (%)
"ΜΕΤΡΗΤΑ"	20
"ΚΑΡΤΑ"	10
"ΔΟΣΕΙΣ"	0

Να γράψετε έναν αλγόριθμο σε Python, ο οποίος:

Γ1. Να διαβάζει με τη χρήση κατάλληλων μηνυμάτων:

α) τη συνολική αξία πριν απο την έκπτωση των προϊόντων που αγόρασε ο πελάτης.

β) τον τρόπο πληρωμής.

(Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών)

Γ2. Να υπολογίζει το ποσό της έκπτωσης ανάλογα με τον τρόπο πληρωμής.

Γ3. Να εμφανίζει το ποσό της έκπτωσης.

Γ4. Να υπολογίζει και να εμφανίζει το τελικό ποσό πληρωμής.

Γ5. Στην περίπτωση που το τελικό ποσό πληρωμής είναι μεγαλύτερο από 200€, να εμφανίζει το μήνυμα «Κερδίσατε Δώρο».

2) Το Υπουργείο Περιβάλλοντος αποφάσισε να παρακολουθήσει για τριάντα (30) ημέρες τα επίπεδα ενός ρύπου στην ατμόσφαιρα, πραγματοποιώντας μία μέτρηση την ημέρα.

Έχουν καθοριστεί τρία επίπεδα μόλυνσης με βάση την τιμή του ρύπου, όπως φαίνεται στον παρακάτω πίνακα:

Τιμές ρύπου	**Επίπεδα μόλυνσης**
έως και 1	«Φυσιολογικό»
πάνω από 1 έως και 2	«Οριακό»
πάνω από 2	«Επικίνδυνο»

Να γράψετε ένα πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. Για κάθε μία από τις τριάντα (30) ημέρες να διαβάζει την τιμή του ρύπου με τη χρήση κατάλληλου μηνύματος (δε χρειάζεται να γίνεται έλεγχος ορθότητας τιμών).

Δ2. Να εμφανίζει για κάθε μέρα το επίπεδο μόλυνσης ανάλογα με την τιμή του ρύπου.

Δ3. Να υπολογίζει και να εμφανίζει το πλήθος των ημερών κατά τη διάρκεια των οποίων η τιμή του ρύπου ξεπέρασε την τιμή 3.

Δ4. Να υπολογίζει και να εμφανίζει τον μέσο όρο των τιμών του ρύπου για το διάστημα των τριάντα (30) ημερών.

ΑΣΚΗΣΗ 1

```
A=input("Δώσε συνολική αξία πριν απο την έκπτωση ")
T=raw_input("Δώσε τρόπο πληρωμής")

if T=="ΜΕΤΡΗΤΑ":
   E=A*20/100

if T=="ΚΑΡΤΑ":
   E=A*10/100

if T=="ΔΟΣΕΙΣ":
   E=0

print "ποσό της έκπτωσης",E

print "τελικό ποσό πληρωμής", A-E

if A-E >200:
   print "Κερδίσατε Δώρο"
```

ΑΣΚΗΣΗ 2

```
c=0
S=0.0

for x in range(30):
   R=input("Δώσε τιμή ρύπου ")

   S=S+R

   if R<=1:
      print "επίπεδο μόλυνσης : Φυσιολογικό"

   if R>1 and R<=2:
      print "επίπεδο μόλυνσης : Οριακό"

   if R>2:
      print "επίπεδο μόλυνσης : Επικίνδυνο"


   if R>3:
      c=c+1

print "μέσος όρος των τιμών του ρύπου ",S/30
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2016

1) Ένα μουσείο τεχνολογίας διοργανώνει στους χώρους του εκπαιδευτικό πρόγραμμα για τα σχολεία της περιοχής. Σε μία διοργάνωση μπορούν να συμμετέχουν έως και 200 μαθητές. Το κόστος ανά μαθητή φαίνεται στον παρακάτω πίνακα:

Αριθμός Μαθητών	**Κόστος ανά μαθητή**
Από 1 έως και 20	5 ευρώ
Από 21 έως και 80	4 ευρώ
Από 81 και πάνω	3 ευρώ

Αν το συνολικό κόστος για το σχολείο είναι μεγαλύτερο από 160 ευρώ, τότε το σχολείο δικαιούται έκπτωση 5%.

Να γραφεί αλγόριθμος σε Python, ο οποίος:

Γ1. Να διαβάζει τον αριθμό των μαθητών που θα συμμετάσχουν σε μία διοργάνωση. Να γίνει έλεγχος ορθότητας τιμών.

Γ 2. Να υπολογίζει και να εμφανίζει το συνολικό κόστος χωρίς την έκπτωση.

Σημειώνεται ότι ο υπολογισμός του συνολικού κόστους δεν είναι κλιμακωτός. Για παράδειγμα, σχολείο με 30 συμμετέχοντες μαθητές θα πληρώσει 30*4=120 ευρώ.

Γ 3. Σε περίπτωση που το σχολείο δικαιούται έκπτωση να υπολογίζει και να εμφανίζει το ποσό της έκπτωσης, καθώς και το τελικό κόστος για το σχολείο. Διαφορετικά, να εμφανίζει το μήνυμα «ΔΕΝ ΔΙΚΑΙΟΥΣΤΕ ΕΚΠΤΩΣΗ».

2) Σε ένα διαγωνισμό χορού συμμετέχουν διαγωνιζόμενοι από όλη τη χώρα. Στην πρώτη φάση του διαγωνισμού κάθε διαγωνιζόμενος βαθμολογείται από τρεις (3) κριτές. Ο διαγωνιζόμενος προκρίνεται στην επόμενη φάση, αν ο μέσος όρος των τριών βαθμολογιών του είναι μεγαλύτερος ή ίσος του επτά (7).

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. Να διαβάζει το επώνυμο κάθε διαγωνιζομένου και τις βαθμολογίες που έλαβε από τους τρεις κριτές. Η επανάληψη συνεχίζεται έως ότου δοθεί για επώνυμο διαγωνιζομένου η τιμή «ΤΕΛΟΣ».

Δ2. Να υπολογίζει το μέσο όρο των βαθμολογιών κάθε διαγωνιζομένου. Εάν ο διαγωνιζόμενος περνάει στην επόμενη φάση, να εμφανίζει το επώνυμο και το μέσο όρο του.

Δ3. Να υπολογίζει και να εμφανίζει το επώνυμο του διαγωνιζομένου με το μεγαλύτερο μέσο όρο. Να θεωρήσετε ότι αυτός ο διαγωνιζόμενος είναι μοναδικός.

Δ4. Να υπολογίζει και να εμφανίζει το πλήθος των διαγωνιζομένων που δεν πέρασαν στην επόμενη φάση.

Υποδείξεις για το Θέμα:

α. Δεν χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

β. Στο διαγωνισμό συμμετέχει τουλάχιστον ένας διαγωνιζόμενος. Να θεωρήσετε ότι δεν υπάρχουν διαγωνιζόμενοι με το ίδιο επώνυμο.

γ. Να θεωρήσετε ότι η βαθμολογία κάθε κριτή είναι μεγαλύτερη ή ίση του ένα (1).

ΑΣΚΗΣΗ 1

```
N=input("Δώσε αριθμό των μαθητών που θα συμμετάσχουν")
while N>200:
   N=input("Δώσε τιμή έως και 200")

if N>=1 and N<=20:
   K=5*N

if N>=21 and N<=80:
   K=4*N

if N>80:
   K=3*N

if N>160:
   E=K*5/100
   print "ποσό της έκπτωσης",E
   print "το τελικό κόστος για το σχολείο",K-E

else:
   print "ΔΕΝ ΔΙΚΑΙΟΥΣΤΕ ΕΚΠΤΩΣΗ"
```

ΑΣΚΗΣΗ 2

```
max1=0
c=0

ON=raw_input("Δώσε επώνυμο")

while ON!="ΤΕΛΟΣ":

   V1=input("Δώσε βαθμό 1")
   V2=input("Δώσε βαθμό 2")
   V3=input("Δώσε βαθμό 3")

   ON=raw_input("Δώσε επώνυμο")

   MO=(V1+V2+V3)/3.0

   if MO>=7:
      print ON
      print MO
   else:
      c=c+1

   if MO>max1:
      max1=MO
      onoma=ON

print "Ο διαγωνιζόμενος με το μεγαλύτερο μέσο όρο",onoma

print "δεν πέρασαν στην επόμενη φάση",c
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2017

1) Σε ένα κεντρικό ΚΤΕΟ προσέρχονται για τεχνικό έλεγχο τριών τύπων οχήματα: Φορτηγά, Επιβατικά και Μοτοσυκλέτες. Οι τακτικοί πελάτες μπορούν να γίνουν μέλη του ΚΤΕΟ και να έχουν έκπτωση στο κόστος ελέγχου. Το κόστος ελέγχου υπολογίζεται σύμφωνα με τ ον παρακάτω πίνακα:

Τύπος οχήματος	**Μέλη ΚΤΕΟ**	**Μη Μέλη ΚΤΕΟ**
Φορτηγά	70 ευρώ	80 ευρώ
Επιβατικά	40 ευρώ	50 ευρώ
Μοτοσυκλέτες	25 ευρώ	30 ευρώ

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο, για μία συγκεκριμένη ημέρα :

Γ1. Να διαβάζει τον τύπο του οχήματος για κάθε όχημα που προσέρχεται καθώς και αν ο πελάτης είναι μέλος του ΚΤΕΟ ή όχι. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως τύπος οχήματος η λέξη « TELOS ».

(Για Φορτηγό θα διαβάζει το « F » , για Επιβατικό το « Ε » και για Μοτοσυκλέτα το « Μ» . Δεν απαιτείται έλεγχος εγκυρότητας εισαγωγής των δεδομένων .)

Γ2. Να υπολογίζει το πλήθος και τις εισπράξεις του ΚΤΕΟ για κάθε τύπο οχήματος και να τα εμφανίζει με κατάλληλα μηνύματα. Για παράδειγμα:

Φορτηγά 10 750 ευρώ

Επιβατικά 20 900 ευρώ

Μοτοσυκλέτες 10 295 ευρώ

(Οι παραπάνω τιμές , όπως και ο τρόπος εμφάνισης -στοίχισης δίνονται ενδεικτικά .)

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος όλων των οχημάτων καθώς και το συνολικό ποσό είσπραξης του ΚΤΕΟ.

Γ4. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα τον αριθμό των μελών του ΚΤΕΟ και των μη μελών που προσήλθαν για τεχνικό έλεγχο την συγκεκριμένη ημέρα.

2) Σε μια Ολυμπιάδα Πληροφορικής συμμετέχουν πενήντα (50) μαθητές. Κάθε μαθητής που συμμετέχει λαμβάνει μια τελική βαθμολογία από 1 έως και 100 ακέραιες μονάδες.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο:

Δ1. Να διαβάζει το ονοματεπώνυμο κάθε μαθητή και τη βαθμολογία που έλαβε. Τα στοιχεία αυτά καταχωρίζονται στις λίστες NAME και VATHMOS αντίστοιχα. Να γίνει έλεγχος ορθότητας ότι δηλαδή η βαθμολογία που καταχωρίζεται είναι από 1 έως και 100.

Δ2. Να υπολογίζει και να εμφανίζει τον μέσο όρο (ΜΟ) της βαθμολογίας όλων των μαθητών.

Δ3. Να εντοπίζει και να εμφανίζει τα ονοματεπώνυμα και τη βαθμολογία όλων των μαθητών των οποίων η τελική βαθμολογία είναι μεγαλύτερη ή ίση από τον μέσο όρο (ΜΟ) .

Δ4. Να εντοπίζει και να εμφανίζει με κατάλληλο μήνυμα την υψηλότερη βαθμολογία και τα ονοματεπώνυμα των μαθητών που έχουν αυτή τη βαθμολογία .

ΑΣΚΗΣΗ 1

```
XF=0
XE=0
XM=0
SF=0
SE=0
SM=0
M1=0
M2=0

T=raw_input("Δώσε τύπο οχήματος")

while T!="TELOS ":

   M=raw_input("είναι μέλος του ΚΤΕΟ ;")

   if T=="F" and M=="Y":
      XF=XF+70
      SF+=1
      M1+=1
   if T=="F" and M=="N":
      XF=XF+80
      SF+=1
      M2+=1

   if T=="E" and M=="Y":
      XE=XE+40
      SE+=1
      M1+=1
   if T=="E" and M=="N":
      XE=XE+50
      SE+=1
      M2+=1

   if T=="M" and M=="Y":
      XM=XM+25
      SM+=1
      M1+=1
   if T=="M" and M=="N":
      XM=XM+30
      SM+=1
      M2+=1

   T=raw_input("Δώσε τύπο οχήματος")

print "Φορτηγά ",SF, XF, "ευρώ"
print "Επιβατικά ",SE, XE, "ευρώ"
print "Μοτοσυκλέτες ",SM, XM, "ευρώ"

print "πλήθος όλων των οχημάτων ", SF+SE+SM
print "συνολικό ποσό είσπραξης ", XF+XE+XM

print "αριθμός μελών ", M1, "αριθμός μη μελών ", M2
```

ΑΣΚΗΣΗ 2

```
NAME=[]
VATHMOS=[]

for x in range(50):
   name=str(raw_input("Δωσε ονομα"))
   vathmos=input("Δωσε βαθμό")
   while vatnos<1 or vathmos>100:
         vathmos=input("Δωσε ΣΩΣΤΟ βαθμό")

   NAME.appens(name)
   VATHMOS.append(vathmos)

S=0.0
for x in VATHMOS:
   S=S+x

MO=S/50

print "μέσος όρος",MO

for x in range(50):
   if VATHMOS[x]>=MO:
      print NAME[x], VATHMOS[x]

max1=VATHMOS[0]
for x in range(1,50,1):
   if VATHMOS[x]> max1:
      max1=VATHMOS[x]

for x in range(50):
   if VATHMOS[x]==max1:
      print NAME[x]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2017

1) Σε μια εθελοντική δράση δενδροφύτευσης συμμετέχουν διάφορα σχολεία.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω:

Γ1. Να διαβάζει, με χρήση μιας δομής επανάληψης, το όνομα του σχολείου και το πλήθος των εθελοντών του. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως όνομα του σχολείου η λέξη «TELOS» (θεωρείστε ότι συμμετέχουν τουλάχιστον 2 σχολεία).

Γ2. Να εμφανίζει το όνομα του σχολείου και το πλήθος των εθελοντών για το σχολείο που έχει τους περισσότερους και για το σχολείο που έχει τους λιγότερους εθελοντές (θεωρείστε ότι ο αριθμός των εθελοντών είναι διαφορετικός και μικρότερος του 100 για κάθε σχολείο).

Γ3. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα το μέσο όρο του αριθμού των εθελοντών όλων των σχολείων.

Γ4. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα το πλήθος των λεωφορείων που θα χρειαστούν για τη μεταφορά των εθελοντών, αν κάθε λεωφορείο διαθέτει 50 θέσεις.

2) Μια εταιρεία κατασκευής υπολογιστών παράγει 20 διαφορετικά μοντέλα υπολογιστών.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω:

Δ1. Να διαβάζει για κάθε μοντέλο το όνομά του και το πλήθος των πωλήσεών του κατά το προηγούμενο έτος. Τα στοιχεία αυτά καταχωρίζονται στις λίστες με ονόματα MODELO και POLISEIS αντίστοιχα.

Δ2. Να υπολογίζει και να εμφανίζει το σύνολο των πωλήσεων όλων των μοντέλων της εταιρείας για το προηγούμενο έτος.

Δ3. Να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της ευθείας ανταλλαγής (φυσαλίδα-bubble sort) τις δύο λίστες σε φθίνουσα σειρά ως προς το πλήθος των πωλήσεων.

Δ4. Να δέχεται το όνομα ενός μοντέλου από το πληκτρολόγιο, να εντοπίζει τις πωλήσεις του και να εμφανίζει το όνομα και το πλήθος των πωλήσεών του. Στη συνέχεια να εμφανίζει τα ονόματα και τις πωλήσεις όλων των μοντέλων που οι πωλήσεις τους είναι μεγαλύτερες ή ίσες από τις πωλήσεις του παραπάνω μοντέλου. Η εμφάνιση να γίνεται σε αύξουσα σειρά ως προς τις πωλήσεις.

ΑΣΚΗΣΗ 1

```
max1=0
min1=100
SumE=0
sxoleia=0

N= raw_input(“ Doste to onoma tou sxoleiou”)

while  N != "TELOS":
   E=int(input(“Dwste aritho ethelontwn”))

   if E > max1:
      max1=E
      MaxName=N

   if E < min1:
      min1=E
      MinName=N

   SumE=SumE + E

   sxoleia= sxoleia + 1

   N = raw_input(“ Dwste to onoma tou sxoleiou”)

print MaxName, max1, MinName, min1

MO=float(SumE)/sxoleia

print " o mesos oros ethelontwn einai:", MO

bus=SumE / 50
if   SumE % 50  !=0 :
   bus+=1

print " ta lewforeia pou tha xreiastoun einai:", bus
```

ΑΣΚΗΣΗ 2

```
MODELO=[]
POLISEIS=[]
for i in range (20):
   M=raw_input("Dwste onoma modelou")
   P=int(input("Dwste upologistes pou poulithikan"))
   MODELO.append (M)
   POLISEIS.append(P)

S=0
for x in POLISEIS:
   S=S+x
print S

for  i in range (1, 20, 1):
   for j in range (19, i-1,-1):
      if POLISEIS[j-1]<POLISEIS[j]:
         POLISEIS[j-1],POLISEIS[j]=POLISEIS[j],POLISEIS[j-1]
         MODELO[j-1], MODELO[j]=MODELO[j],MODELO[j-1]

Onoma=raw_input("Dwste onoma modelou")

for x in range (20):
   if MODELO[x]==Onoma:
      POL=POLISEIS[x]
      print MODELO[x],POLISEIS[x]

for x in range (19, -1,-1):
   if   POLISEIS[x] >= POL:
      print MODELO[x], POLISEIS[x]
```

ΥΠΟΛΕΙΠΟΜΕΝΕΣ ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2017

1) Σε μια εξέταση του μαθήματος της Αγγλικής Γλώσσας εξετάζονται πενήντα (50) μαθητές προφορικά και γραπτά. Οι μαθητές βαθμολογούνται από το 0 .0 έως και το 20.0 σε κάθε εξέταση (προφορικά, γραπτά).

Να γράψετε ένα πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω :

Γ1. Σε δομή επανάληψης να διαβάζει το ονοματεπώνυμο , την προφορική και τη γραπτή βαθμολογία κάθε μαθητή. Δεν απαιτείται έλεγχος ορθότητας εισαγωγής τιμών.

Γ2 . Να εμφανίζει τα ονοματεπώνυμα των μαθητών που έχουν άθροισμα προφορικής και γραπτής βαθμολογίας μεγαλύτερο από το 19.5.

Γ3 . Να υπολογίζει και να εμφανίζει το πλήθος των μαθητών που η γραπτή βαθμολογία τους είναι μεγαλύτερη από την προφορική τους .

Γ4 . Να υπολογίζει και να εμφανίζει τον μέσο όρο της γραπτής βαθμολογίας και τον μέσο όρο της προφορικής βαθμολογίας όλων των μαθητών.

2) Το Υπουργείο Πολιτισμού διατηρεί στατιστικά στοιχεία για το θέατρο της Αρχαίας Επιδαύρου σχετικά με τον τίτλο κάθε παράστασης και το πλήθος των θεατών που την παρακολούθησαν (κάθε παράσταση παρουσιάζεται μόνο μία φορά και υπάρχει τουλάχιστον μία παράσταση) .

Να γράψετε ένα πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω :

Δ1. Να διαβάζει τον τίτλο κάθε παράστασης και το πλήθος των θεατών που την παρακολούθησαν. Τα στοιχεία αυτά να καταχωρίζονται στις λίστες με ονόματα PAR και S _ P αντίστοιχα. Να γίνεται έλεγχος ορθότητας για το πλήθος των θεατών που εισάγεται έτσι ώστε να είναι θετικός αριθμός. Η εισαγωγή των στοιχείων θα τερματίζεται όταν δοθεί ως τίτλος παράστασης η λέξη « TELOS » .

Δ2. Να εντοπίζει και να εμφανίζει τον τίτλο της παράστασης με το μέγιστο πλήθος θεατών. Να θεωρήσετε ότι υπάρχει μία μόνο τέτοια παράσταση.

Δ3. Να υπολογίζει και εμφανίζει τον μέσο όρο των θεατών όλων τ ων παραστάσεων .

Δ4. Κάθε παράσταση με πλήθος θεατών μεγαλύτερο ή ίσο από 1000 άτομα επιδοτείται με 10000€, ενώ κάθε παράσταση με πλήθος θεατών μικρότερο των 1000 ατόμων επιδοτείται με 5000€. Να υπολογίσετε και να εμφανίσετε το συνολικό ποσό της επιδότησης που θα διαθέσει το Υπουργείο Πολιτισμού.

ΑΣΚΗΣΗ 1

```
S1=0
S2=0
c=0
for x in range(50):
   name=str(raw_input("Δωσε ονομα"))

   vathmos1=float(input("Δωσε γραπτο βαθμό"))
   while vatnos1<0.0 or vathmos1>20.0:
        vathmos1=float(input("Δωσε ΣΩΣΤΟ βαθμό"))

   vathmos2=float(input("Δωσε προφορικό βαθμό"))
   while vatnos2<0.0 or vathmos2>20.0:
        vathmos2=float(input("Δωσε ΣΩΣΤΟ βαθμό"))

   if vathmos1 + vathmos2 >19.5:
      print name

   if vathmos1 > vathmos2 :
      c=c+1

   S1=S1+vathmos1
   S2=S2+vathmos2


print "πλήθος με γραπτή βαθμολογία  μεγαλύτερη από την προφορική", c

print "μέσο όρο γραπτής βαθμολογίας ",S1/20

print "μέσο όρο προφορικής βαθμολογίας ",S2/20
```

ΑΣΚΗΣΗ 2

```
PAR=[]
S_P=[]

T=str(raw_input("Δωσε τίτλο παράστασης"))

while T!="TELOS":
   P=input("Δώσε πλήθος θεατών ")
   while P<=0:
      P=input("Δώσε ΣΩΣΤΟ πλήθος θεατών ")

   PAR.append(T)
   S_P.append(P)

EP=0
S=S_P[0]
max1=S_P[0]

for x in range(1,len(PAR)):
   if S_P[x]>max1:
      max1=S_P[x]
      titlos=PAS[x]

   S=S+S_P[x]

   if S_P[x]>=1000:
      EP=EP+10000
   else:
      EP=EP+5000

print "παράστασης με το μέγιστο πλήθος θεατών",titlos

print "μέσο όρο θεατών ", S/float(len(S_P))

print "συνολικό ποσό επιδότησης ", EP
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2018

1) Διαφημιστική εταιρεία επιθυμεί να παραγγείλει κεφαλαία γράμματα του ελληνικού αλφαβήτου για τη δημιουργία επιγραφών. Για το λόγο αυτό κάνει στατιστική εκτίμηση της συχνότητας εμφάνισης των γραμμάτων χρησιμοποιώντας τη λίστα GRAM η οποία περιέχει στη σειρά τα κεφαλαία ελληνικά γράμματα, δηλαδή GRAM =[' Α ' , ' Β ' , ' Γ ' , . . . , ' Ψ' , ' Ω '] .

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο:

Γ1. Να διαβάζει από το πληκτρολόγιο διαδοχικά δύο επιγραφές με κεφαλαία ελληνικά γράμματα και να τις συνενώνει στη μεταβλητή epig .

Γ2. Να δημιουργεί μια κενή λίστα με όνομα SUMA και στη συνέχεια με μια επαναληπτική διαδικασία να καταχωρίζει σε αυτή 24 στοιχεία με τιμή μηδέν (0). Η λίστα SUMA θα χρησιμοποιηθεί στα επόμενα ερωτήματα, για την αποθήκευση του αριθμού που δείχνει πόσες φορές υπάρχει κάθε γράμμα (συχνότητα εμφάνισης) στη μεταβλητή epig . Κάθε θέση της λίστας SUMA αντιστοιχεί , με την ίδια σειρά, σε ένα γράμμα της λίστας GRAM .

Γ3 . Να υπολογίζει τη συχνότητα εμφάνισης κάθε γράμματος της λίστας GRAM που περιέχεται στη μεταβλητή epig και να ενημερώνει την αντίστοιχη θέση της λίστας SUMA με την τιμή αυτή.

Γ4. α) Να εμφανίζει ποια γράμματα πρέπει να παραγγελθούν και σε ποια ποσότητα.

β) Να υπολογίζει και να εμφανίζει το πλήθος των γραμμάτων που έχουν μηδενικό πλήθος εμφανίσεων και δεν θα παραγγελθούν .

Σημείωση : α) Η λίστα GRAM θα πρέπει να οριστεί στο πρόγραμμα που θα αναπτύξετε.

β) Δεν απαιτούνται έλεγχοι ορθότητας δεδομένων .

2) Δίνεται λίστα Α η οποία περιέχει ονόματα πόλεων και τη μέγιστη θερμοκρασία τους σε μία συγκεκριμένη ημέρα . Η λίστα έχει την παρακάτω δομή (ενδεικτικά):

Πάτρα, 25, Λάρισα, 27

Συνεπώς τα στοιχεία που βρίσκονται σε άρτιες θέσεις περιέχουν ονόματα πόλεων και τα στοιχεία που βρίσκονται σε περιττές θέσεις περιέχουν σε ακέραιο αριθμό τη μέγιστη θερμοκρασία της πόλης .

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο:

Δ1. Να διαβάζει την λίστα Α και ανάλογα αν είναι άρτια ή περιττή θέση, να καταχωρεί τα ονόματα των πόλεων σε μία λίστα POL και τις αντίστοιχες θερμοκρασίες τους σε μία λίστα THER.

Δ2. Να υπολογίζει και να εμφανίζει το μέσο όρο των θερμοκρασιών όλων των πόλεων.

Δ 3 . Να ταξινομεί τη λίστα THER με χρήση του αλγορίθμου ταξινόμησης της ευθείας ανταλλαγής (φυσαλίδα – bubblesort) σε φθίνουσα σειρά ως προς τις θερμοκρασίες αναδιατάσσοντας συγχρόνως τη λίστα POL έτσι ώστε να διατηρείται η αντιστοίχιση πόλεων - θερμοκρασιών .

Δ4 . Θεωρώντας ότι μπορεί να υπάρχουν περισσότερες από μία πόλεις με την ίδια θερμοκρασία, να εμφανίζει την υψηλότερη θερμοκρασία που έχει καταχωρηθεί και τα ονόματα των πόλεων που έχουν αυτή τη θερμοκρασία.

ΑΣΚΗΣΗ 1

```
GRAM = ['Α','Β','Γ','Δ','Ε','Ζ','Η','Θ','Ι','Κ','Λ','Μ','Ν','Ξ','Ο','Π','Ρ','Σ','Τ','Υ','Φ','Χ','Ψ','Ω']
epig1 = raw_input('Δώσε την 1η επιγραφή:')
epig2 = raw_input('Δώσε την 2η επιγραφή:')
epig = epig1 + epig2

SUMA = []
for i in range(24):
   SUMA = SUMA + [0]

for letter in epig:
   for i in range(24):
      if letter == GRAM[i]:
         SUMA[i] = SUMA[i] + 1
pl = 0
for i in range(24):
   if SUMA[i]>0:
      print GRAM[i],':',SUMA[i]
   else:
      pl = pl + 1
print "Πλήθος γραμμάτων με μηδενικό πλήθος εμφανίσεων:",pl
```

ΑΣΚΗΣΗ 2

```
POL = []
THER = []
count = 0

i = 0
for x in A:
   i = i + 1
   if  i % 2 == 0 :
      POL.append(x)
   else:
      THER.append(float(x))

S = 0.0
for  x  in  THER:
   S = S + x
print "MO:", S / len(THER)

N=len(THER)
for i in range(1,N, 1):
   for j in range(N-1,i-1,-1):
      if   THER[j] > THER[j-1]:
         THER[j], THER[j-1] = THER[j-1] , THER[j]
         POL [j], POL [j-1] = POL [j-1] , POL [j]

print "Υψηλότερη θερμοκρασία:", THER[0]

for x   in  range(N):
     if THER[x] == THER[0]:
         print POL[x]
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2019

1) Σε έναν αγώνα ταχύτητας αυτοκινήτων συμμετέχουν στον προκριματικό 35 οδηγοί με τα αυτοκίνητα τους. Στον τελικό αγώνα θα συμμετάσχουν όσοι οδηγοί σημειώσουν επίδοση μικρότερη ή ίση από 180 δευτερόλεπτα που αποτελεί το όριο πρόκρισης. Κάθε οδηγός έχει μέχρι τέσσερις (4) προσπάθειες για να πετύχει το όριο πρόκρισης. Αν πετύχει σε μία προσπάθεια, σταματά και δεν συνεχίζει τις υπόλοιπες προσπάθειες.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Για κάθε οδηγό να διαβάζει το όνομά του και διαδοχικά τον χρόνο των προσπαθειών του μέχρι να πετύχει την κατάλληλη επίδοση ή να συμπληρωθεί ο αριθμός των προσπαθειών που δικαιούται.

Γ2. Για κάθε οδηγό να εμφανίζει το όνομά του και αν προκρίθηκε τον χρόνο πρόκρισής του, διαφορετικά να εμφανίζει το μήνυμα “ΜΗ ΣΥΜΜΕΤΟΧΗ”.

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος των προκριθέντων οδηγών, καθώς και το μέσο όρο των χρόνων πρόκρισης που πέτυχαν. Υποθέστε ότι υπάρχει τουλάχιστον ένας.

Γ4. Να βρίσκει και να εμφανίζει το όνομα του οδηγού με το μικρότερο χρόνο πρόκρισης και τον χρόνο αυτό (Υποθέστε ότι είναι μοναδικός).

2) Σε μία αποθήκη σταθμού τρένων υπάρχει ένας πεπερασμένος αριθμός από κιβώτια εμπορευμάτων. Τα κιβώτια πρόκειται να φορτωθούν σε άδεια βαγόνια ενός τρένου. Κάθε βαγόνι έχει όριο χωρητικότητας **2000 λίτρα**. Η φόρτωση ακολουθεί την παρακάτω διαδικασία:

Σε κάθε βαγόνι φορτώνονται διαδοχικά κιβώτια με προκαθορισμένη σειρά, μέχρι να συμπληρωθεί το όριο χωρητικότητάς του. Ένα κιβώτιο φορτώνεται στο βαγόνι μόνο εάν ο όγκος του μαζί με τον όγκο των ήδη φορτωμένων κιβωτίων δεν ξεπερνούν το όριο χωρητικότητας του βαγονιού. Διαφορετικά η φόρτωση του βαγονιού ολοκληρώνεται και το κιβώτιο φορτώνεται στο επόμενο βαγόνι. Αν δεν υπάρχει άλλο διαθέσιμο βαγόνι, το κιβώτιο παραμένει στην αποθήκη.

Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να τελειώσουν τα κιβώτια ή να μην υπάρχει άλλο διαθέσιμο βαγόνι.

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. α. Να διαβάζει τον αριθμό των διαθέσιμων βαγονιών του τρένου.

β. Να διαβάζει τον όγκο κάθε κιβωτίου της αποθήκης σε λίτρα και να το εισάγει σε μία λίστα με όνομα QUE, έως ότου εισαχθεί ως όγκος κιβωτίου ο αριθμός μηδέν (0).

Δ2. Να υπολογίζει και να εμφανίζει, με κατάλληλο μήνυμα, για κάθε βαγόνι που χρησιμοποιήθηκε, το πλήθος και τον συνολικό όγκο των κιβωτίων που περιέχει. Κάθε κιβώτιο που φορτώνεται αφαιρείται από τη λίστα

Δ3. Να υπολογίζει τον αριθμό των βαγονιών που χρησιμοποιήθηκαν. Αν φορτώθηκαν όλα τα κιβώτια, να υπολογίζει το συνολικό όγκο τους και να εμφανίζει με κατάλληλο μήνυμα, τον αριθμό των βαγονιών που χρησιμοποιήθηκαν και τον συνολικό όγκο. Διαφορετικά, αν δεν φορτώθηκαν όλα, να υπολογίζει και να εμφανίζει το πλήθος των κιβωτίων που παρέμειναν στη λίστα (αποθήκη) καθώς και τον συνολικό όγκο σε όλα τα βαγόνια που δεν αξιοποιήθηκε.

Θεωρείστε ότι κανένα κιβώτιο δεν έχει όγκο μεγαλύτερο από 2.000 λίτρα

ΑΣΚΗΣΗ 1

```
pl = 0
Sum = 0.0
Min = 181
#Γ1
for i in range(35):
   onoma = raw_input("Δώσε όνομα:")
   epidosi = input("Δώσε επίδοση:")
   pr = 1
   while  epidosi > 180   and  pr < 4:
      epidosi = input("Δώσε επίδοση:")
      pr = pr + 1

   #Γ2
   if   epidosi <= 180:
      print onoma, epidosi
      #Γ3
      pl = pl + 1
      Sum = Sum + epidosi
   else:
      print onoma,"ΜΗ ΣΥΜΜΕΤΟΧΗ"
#Γ4
   if   epidosi < Min:
      Min = epidosi
      onomaMin = onoma
#Γ3
print pl, Sum/pl

#Γ4
print onomaMin, Min
```

ΑΣΚΗΣΗ 2

```
QUE = []
#Δ1
N = input("Δώσε αριθμό βαγονιών:")

v = input("Δώσε τον όγκο του κιβωτίου:")
while v != 0:
   QUE.append(v)
   v = input("Δώσε τον όγκο του κιβωτίου:")

#Δ2
Q1 = []
Q2 = []
i = 1
SumV = 0
while  i <= N and QUE != []:
   pl = 0
   Sum = 0
   v = QUE[0]
   while   Sum + v <= 2000   and QUE != []:
      Sum = Sum + v
      QUE.pop(0)

      pl = pl + 1
      if QUE != []:
         v = QUE[0]
   Q1.append(pl)
   Q2.append(Sum)
   SumV = SumV + Sum
   i = i + 1


for   i in range(len(Q1)):
     print "Το βαγόνι",i+1,"είχε",Q1[i],"κιβώτια και συνολικό όγκο",Q2[i]

#Δ3
if QUE == []:
   print "Χρησιμοποιήθηκαν",len(Q1),"βαγόνια με συνολικό όγκο",SumV
else:
   print "Παρέμειναν στην αποθήκη",len(QUE),"κιβώτια"
   print "Ο συνολικός όγκος των βαγονιών που δεν αξιοποιήθηκε είναι",2000*N  - SumV
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2020

1) Σε μία παράσταση υπαίθριου δημοτικού θεάτρου το εισιτήριο εισόδου είναι για τους ενήλικες δέκα (10) ευρώ και για τα παιδιά πέντε (5) ευρώ. Η χωρητικότητα του θεάτρου είναι πεντακόσια (500) άτομα.

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο να πραγματοποιεί τα παρακάτω :

Γ1.

α) Για κάθε άτομο ή παρέα εισερχόμενων θεατών, να εμφανίζει τις διαθέσιμες ελεύθερες θέσεις του θεάτρου και να διαβάζει από το πληκτρολόγιο το πλήθος των ενηλίκων και ακολούθως το πλήθος των παιδιών της παρέας .

β) Να καλεί τη συνάρτηση EISITIRIO () , η οποία υπολογίζει το συνολικό κόστος των εισιτηρίων, και στη συνέχεια το πρόγραμμα να εμφανίζει το κόστος αυτό. Η λειτουργία της συνάρτησης αυτής περιγράφεται στο ερώτημα Γ3.

γ) Η διαδικασία της εισόδου θεατών να τερματίζεται, όταν εισαχθεί ο αριθμός μείον ένα (- 1) ως πλήθος ενηλίκων θεατών μιας παρέας. Ο αριθμός μείον ένα (- 1) σημαίνει ότι είτε οι διαθέσιμες θέσεις δεν επαρκούν είτε δεν υπάρχουν άλλοι θεατές που επιθυμούν να εισέλθουν. Στην περίπτωση αυτή, δεν εισάγεται αριθμός παιδιών. Θεωρήστε ότι υπάρχει τουλάχιστον ένας θεατής και σε κάθε παρέα υπάρχει τουλάχιστον ένας ενήλικας.

Γ2. Να υπολογίζει και να εμφανίζει:

α) Τα συνολικά έσοδα του θεάτρου.

β) Το ποσοστό των παιδιών στο σύνολο των θεατών

Γ3. Να υλοποιήσετε τη συνάρτηση EISITIRIO () , η οποία δέχεται το πλήθος των ενηλίκων και το πλήθος των παιδιών μιας παρέας και επιστρέφει το συνολικό κόστος των εισιτηρίων.

2) Ένας οργανισμός διεξάγει διαγωνισμό για την πρόσληψη τριών (3) υπαλλήλων, στον οποίο προσήλθαν είκοσι (20) υποψήφιοι. Οι υποψήφιοι , μεταξύ άλλων, θα πρέπει να συμπληρώσουν ένα τεστ που αποτελείται από δεκαπέντε (15) ερωτήσεις πολλαπλής επιλογής . Κάθε ερώτηση έχει πέντε (5) δυνατές απαντήσεις , οι οποίες χαρακτηρίζονται από τα γράμματα α, β, γ, δ και ε. Μία από τις απαντήσεις α, β, γ και δ είναι η σωστή, ενώ η απάντηση " ε " σημαίνει «Δεν γνωρίζω». Ο υποψήφιο ς για κάθε ερώτηση έχει μία και μόνο επιλογή. Αν η επιλογή αντιστοιχεί στη σωστή απάντηση βαθμολογείται με τρεις (3) μονάδες, αν είναι λανθασμένη, αφαιρείται μία (1) μονάδα και αν έχει επιλεγεί το " ε " , δεν βαθμολογείται [μηδέν (0) μονάδες]. Οι σωστές απαντήσεις βρίσκονται στη λίστα LI [" α ", " δ ", " γ ", " β ", " δ ", " γ ", " β " , " α ", " δ ", " γ ", " β ", " δ ", " γ ", " β " , " α "] .

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο να πραγματοποιεί τα παρακάτω :

Δ1. Για κάθε υποψήφιο:

α) Να διαβάζει από το πληκτρολόγιο το όνομά του και να το εισάγει στη λίστα ΟΝ [ ].

β) Να διαβάζει από το πληκτρολόγιο διαδοχικά τις δεκαπέντε (15) απαντήσεις που έδωσε και να υπολογίζει τη συνολική βαθμολογία του, σύμφωνα με την περιγραφόμενη διαδικασία βαθμολόγησης και χρήση της λίστα ς LI [ ] . Η συνολική βαθμολογία του εισάγεται στη λίστα SV [ ]

Δ2. Να εμφανίζει τα ονόματα των υποψηφίων που είχαν βαθμολογία μεγαλύτερη ή ίση από τον μέσο όρο όλων των βαθμολογιών.

Δ3. Να εμφανίζει τα ονόματα των τριών (3) υποψηφίων που είχαν την υψηλότερη βαθμολογία. (Θεωρήστε ότι οι βαθμολογίες είναι διαφορετικές μεταξύ τους).

ΑΣΚΗΣΗ 1

```
def Eisitirio(E,P):
   return E*10+P*5

SP=0
T=500
S=0

E=input("Δώσε αριθμό ενηλίκων")
while E!=-1:
   P=input("Δώσε αριθμό παιδιών")

   Sp=Sp+P
   Xr=Eisitirio(E,P)

   print "Διαθέσιμες ελεύθερες θέσεις", T
   print "Κόστος εισιτηρίων ",Xr

   S=S+Xr
   T=T-E-P

   E=input("Δώσε αριθμό ενηλίκων ή -1 ")

print "Τα συνολικά έσοδα του θεάτρου", S

print "Το ποσοστό των παιδιών στο σύνολο των θεατών", 100*Sp/(500-T)
```

ΑΣΚΗΣΗ 2

```
ON=[]
SV=[]

for x in range(20):
   on=raw-input("Δώσε όνομα")
   ON.append(on)
   S=0
   for y in range(15):
      a=raw_input("Δώσε απάντηση")
      if a=="ε":
         m=0
      elif a==LI[y]:
         m=3
      else:
         m=-1
      S=S+m
   SV.append(S)

S=0.0
for x in SV:
   S=S+x

MO=S/20

for x in range(20):
   if SV[x]>=MO:
      print ON[x]


N=20
for i in range(1,N,1):
   for j in range (N-1,i-1,-1):
      if SV[j]>SV[j-1]:
         SV[j],SV[j-1]=SV[j-1],Sv[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]

print ON[0], ON[1], ON[2]
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ

ΓΕΛ

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2001

1)Σε ένα πρόγραμμα περιβαλλοντικής εκπαίδευσηςσυμμετέχουν20 σχολεία. Στα πλαίσια αυτού τουπρογράμματος, εθελοντές μαθητές των σχολείων, πουσυμμετέχουν στο πρόγραμμα, μαζεύουν ποσότητες τριώνυλικών(γυαλί, χαρτί και αλουμίνιο).

Να αναπτύξετε έναν αλγόριθμο, ο οποίος:

α. να διαβάζει τις ποσότητες σε κιλά των παραπάνω υλικώνπου μάζεψαν οι μαθητές σε κάθε σχολείο

β. να υπολογίζει και να τυπώνει τη συνολική ποσότητα σε κιλά του κάθευλικού που μάζεψαν οι μαθητές σε όλα τα σχολεία

γ. αν η συνολική ποσότητα του χαρτιού που μαζεύτηκε από όλα τα σχολεία είναι λιγότερη των1000 κιλών, να εμφανίζεται το μήνυμα«Συγχαρητήρια». Αν η ποσότητα είναι από1000 κιλά και πάνω, αλλά λιγότερο από2000, να εμφανίζεται το μήνυμα«Δίνεται έπαινος» και τέλος αν η ποσότητα είναι από2000 κιλά και πάνω να εμφανίζεται το μήνυμα «Δίνεται βραβείο».

Παρατήρηση: Να θεωρήσετε ότι όλες οι ποσότητες είναι θετικοί αριθμοί.

ΑΣΚΗΣΗ 1

```
SG=0
SP=0
SA=0

for x in range(20):
   g=input("Δώσε ποσότητα γυαλιού ")
p=input("Δώσε ποσότητα χαρτιού ")
a=input("Δώσε ποσότητα αλουμίνιου ")
SG=SG+g
   SP=SP+p
   SA=SA+a

print "συνολική ποσότητα γυαλιού",SG
print "συνολική ποσότητα χαρτιού",SP
print "συνολική ποσότητα αλουμίνιου",SA

ifSG>=0 andSG<1000:
print "Συγχαρητήρια"
if SG>=1000 and SG<2000:
   print "Δίνεται έπαινος"
ifSG>=2000 :
print "Δίνεται βραβείο"
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣΕΣΠΕΡΙΝΟΥ ΓΕΛ 2001

1) Ένας μαθητής που τελείωσε το λύκειο με άριστα, ζήτησε σαν δώρο απ' τους γονείς του να του πάρουν ένα Η/Υ αξίας 1500 €. Οι γονείς του είπαν ότι θα του δίνουν σταδιακά το ποσό, δίνοντάς του κάθε εβδομάδα, ποσό διπλάσιο από την προηγούμενη, ξεκινώντας από 15€.

Να γράψετε πρόγραμμα σε Pythonπου

α) υπολογίζει και εμφανίζει μετά από πόσες βδομάδες θα μπορέσει να πάρει ο μαθητής τον Η/Υ και,

β) να υπολογίζει, να ελέγχει και να τυπώνει πιθανό περίσσευμα χρημάτων. (3ο θέμα πανελληνίων 2001 – Εσπερινά λύκεια)

2) Σε κάποια εξεταστική δοκιμασία ένα γραπτό αξιολογείται από δύο βαθμολογητές στη βαθμολογικήκλίμακα[0, 100].

Αν η διαφορά μεταξύ των βαθμολογιών του α΄ καιτου β΄ βαθμολογητή είναι μικρότερη ή ίση των20 μονάδων της παραπάνω κλίμακας, ο τελικός βαθμόςείναι ο μέσος όρος των δύο βαθμολογιών.

Αν η διαφορά μεταξύ των βαθμολογιών του α΄ καιτου β΄ βαθμολογητή είναι μεγαλύτερη από20 μονάδες, το γραπτό δίνεται για αναβαθμολόγηση σετρίτο βαθμολογητή. Ο τελικός βαθμός του γραπτούπροκύπτει τότε από τον μέσο όρο των τριώνβαθμολογιών.

Να αναπτύξετε αλγόριθμο σε Python ο οποίος, αφού ελέγξειτην εγκυρότητα των βαθμών στην βαθμολογικήκλίμακα[0, 100], να υλοποιεί την παραπάνωδιαδικασία εξαγωγής τελικού βαθμού και ναεμφανίζει τον τελικό βαθμό του γραπτού στηνεικοσαβάθμια κλίμακα.

Παρατήρηση: Να θεωρήσετε ότι όλες οι ποσότητεςεκφράζονται ως πραγματικοί αριθμοί.

ΑΣΚΗΣΗ 1

```
P=15
C=1500
M=0
while P<= C :
   C=C-P
   P=P*2
   M=M+1

print "Evdomades: ", M
print "Perisevma xrimaton:", P-C
```

ΑΣΚΗΣΗ 2

```
A=input("Δώσε βαθμό 1")
while A<0 or A>100:
   A=input("Δώσε ΣΩΣΤΟ βαθμό 1")

B=input("Δώσε βαθμό 2")
while B<0 or B>100:
   B=input("Δώσε ΣΩΣΤΟ βαθμό 2")

if abs(A-B)<=20:
   print "τελικός βαθμός", 0.2*(A+B)/2.0

else:
   C=input("Δώσε βαθμό 3")
   while C<0 or C>100:
      C=input("Δώσε ΣΩΣΤΟ βαθμό 1")
print "τελικός βαθμός", 0.2*(A+B+C)/3.0
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2002

1) Με το νέο σύστημα πληρωμής των διοδίων, οι οδηγοί των τροχοφόρων έχουν τη δυνατότητα να πληρώνουν το αντίτιμο των διοδίων με ειδική μαγνητική κάρτα. Υποθέστε ότι υπάρχει μηχάνημα το οποίο διαθέτει είσοδο για την κάρτα και φωτοκύτταρο. Το μηχάνημα διαβάζει από την κάρτα το υπόλοιπο των χρημάτων και το αποθηκεύει σε μία μεταβλητή Υ και, με το φωτοκύτταρο, αναγνωρίζει τον τύπο του τροχοφόρου και το αποθηκεύει σε μία μεταβλητή Τ. Υπάρχουν τρεις τύποι τροχοφόρων: δίκυκλα (Δ), επιβατικά (Ε) και φορτηγά (Φ), με αντίτιμο διοδίων 1, 2 και 3 ευρώ αντίστοιχα.

Να αναπτύξετε αλγόριθμο σε Python, ο οποίος:

α. ελέγχει τον τύπο του τροχοφόρου και εκχωρεί στη μεταβλητή Α το αντίτιμο των διοδίων, ανάλογα με τον τύπο του τροχοφόρου

β. ελέγχει την πληρωμή των διοδίων με τον παρακάτω τρόπο.

Αν το υπόλοιπο της κάρτας επαρκεί για την πληρωμή του αντιτίμου των διοδίων, αφαιρεί το ποσό αυτό από την κάρτα. Αν η κάρτα δεν έχει υπόλοιπο, το μηχάνημα ειδοποιεί με μήνυμα για το ποσό που πρέπει να πληρωθεί. Αν το υπόλοιπο δεν επαρκεί, μηδενίζεται η κάρτα και δίνεται με μήνυμα το ποσό που απομένει να πληρωθεί.

2) Μια εταιρεία αποθηκεύει είκοσι (20) προϊόντα σε δέκα (10) αποθήκες. Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού Python, το οποίο:

α. εισάγει σε λίστα ΟΝτα ονόματα των είκοσι προϊόντων

β. εισάγει σε λίσταΡτην πληροφορία που αφορά στην παρουσία ενός προϊόντος σε μια αποθήκη (καταχωρούμε την τιμή 1 στην περίπτωση που υπάρχει το προϊόν στην αποθήκη και την τιμή 0, αν το προϊόν δεν υπάρχει στην αποθήκη).

γ. υπολογίζει σε πόσες αποθήκες βρίσκεται το κάθε προϊόν

δ. τυπώνει το όνομα κάθε προϊόντος και το πλήθος των αποθηκών στις οποίες υπάρχει το προϊόν.

ΑΣΚΗΣΗ 1

```
T=raw_input("Δώσε τύπο οχήματος")
Y=input("Δώσε υπόλοιπο κάρτας")

if T=="Δ":
   A=1

if T=="Ε":
   A=2

if T=="Φ":
   A=3

if Y>=A:
   Y=Y-A

if Y==0:
   print A

if Y>0 and Y<A:
   Y=0
   print A
```

ΑΣΚΗΣΗ 2

```
ON=[]
P=[]


for x in range(10):
   on=str(raw_input("Δωσε ονομα"))
   ON.append(on)

for x in rnage(10):
   T=[]
   for y in range(20):
      print "προιόν ", y+1
t=input("Δώσε τιμή 1 ή 0")
T.append(t)
   P.append(T)

for x in range(20):
   c=0
   for y in range(10):
      if P[y] [x] ==1:
         c=c+1
print "Το προιον ",ON[x], "βρίσκεται σε ",c, "αποθήκες"
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣΕΣΠΕΡΙΝΟΥ ΓΕΛ 2002

1) Δίνονται η έκταση, ο πληθυσμός και το όνομα καθεμιάς από τις 15 χώρες της Ευρωπαϊκής Ένωσης.

Να αναπτύξετεαλγόριθμο σε Python που

α) θα διαβάζει τα παραπάνω δεδομένα,

β) θα εμφανίζει τη χώρα με τη μεγαλύτερη έκταση,

γ) θα εμφανίζει τη χώρα με το μικρότερο πληθυσμό και

δ) θα εμφανίζει το μέσο όρο του πληθυσμού των15 χωρώντης Ευρωπαϊκής Ένωσης.

2)Στο πλαίσιο προγράμματος προληπτικής ιατρικής για τηναντιμετώπιση του νεανικού διαβήτη έγιναν αιματολογικέςεξετάσεις στους 90 μαθητές(αγόρια και κορίτσια) ενός Γυμνασίου.

Για κάθε παιδί καταχωρίστηκαν τα ακόλουθα στοιχεία:

1. **ονοματεπώνυμο μαθητή**
2. **κωδικός φύλου ("Α" για τα αγόρια και "Κ" για τα κορίτσια)**
3. **περιεκτικότητα σακχάρου στο αίμα.**

Οι φυσιολογικές τιμές σακχάρου στο αίμα κυμαίνονται από70 έως110 mg/dl (συμπεριλαμβανομένων και των ακραίωντιμών).

Να αναπτύξετε αλγόριθμο σε Python που

α) θα διαβάζει τα παραπάνω στοιχεία(ονοματεπώνυμο, φύλο, περιεκτικότητα σακχάρου στο αίμα) και θα ελέγχειτην αξιόπιστη καταχώρισή τους (δηλαδή το φύλο να είναι μόνο"Α" ή"Κ" και η περιεκτικότητα σακχάρου στο αίμανα είναι θετικός αριθμός),

β) θα εμφανίζει για κάθε παιδί του οποίου η περιεκτικότητασακχάρου στο αίμα είναι εκτός των φυσιολογικών τιμών, το ονοματεπώνυμο, το φύλο και την περιεκτικότητα τουσακχάρου,

γ) θα εμφανίζει το συνολικό αριθμό των αγοριών τωνοποίων η περιεκτικότητα σακχάρου στο αίμα δεν είναιφυσιολογική και

δ) θα εμφανίζει το συνολικό αριθμό των κοριτσιών τωνοποίων η περιεκτικότητα σακχάρου στο αίμα δεν είναιφυσιολογική.

ΑΣΚΗΣΗ 1

```
maxE=0
S=0.0

for x in range (15):
   on=raw_input("Δώσε ονομα")
   p=input("Δώσε πληθυσμός ")
e=input("Δώσε έκταση")

if maxE<e:
      e=maxE
      onE=on
   if x==0:
      minP=p
      onP=on
   else:
      if minP>p:
         minP=p
         onP=on

S=S+p

print "χώρα με τη μεγαλύτερη έκταση",onE

print "χώρα με το μικρότερο πληθυσμό",onP

print "μέσο όρο του πληθυσμού", S/15
```

ΑΣΚΗΣΗ 2

```
c1=0
c2=0
for x in range(90):
   on=raw_input("Δώσε ονομα")

   f=raw_input("Δώσε κωδικό φύλου")
while f not in ["A", "K"]:
      f=raw_input("Δώσε ΣΩΣΤΟ κωδικό φύλου")

p=input("Δώσε περιεκτικότητα σακχάρου")
whilep<0:
p=input("Δώσε ΣΩΣΤΗ περιεκτικότητα σακχάρου")

if p<70 or p>110:
      print on, f, p

      if f=="A":
         c1=c1+1
iff=="K":
c2=c2+1

print "αριθμό αγοριών οχι φυσιολογική περιεκτικότητα",c1
print "αριθμό κοριτσιών οχι φυσιολογική περιεκτικότητα",c2
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2003

1) Ο Δείκτης Μάζας του ανθρώπινου σώματος (ΔΜΣ) υπολογίζεται από το Βάρος (Β) και το Ύψος (Υ) με τον τύπο ΔΜΣ = Β/Υ*2 . Ο τύπος αυτός ισχύει για ανθρώπους σε ηλικία άνω των 18 ετών. Ένας άνθρωπος ανάλογα με την τιμή του ΔΜΣ χαρακτηρίζεται σύμφωνα με τον παρακάτω πίνακα:

ΔΜΣ < 18.5	Αδύνατο
18,5 <= ΔΜΣ < 25	Κανονικό
25 <= ΔΜΣ < 30	Βαρύ
30 <= ΔΜΣ	Υπέρβαρο

Να γράψετε αλγόριθμο ο οποίος :

Α. Να διαβάζει την ηλικία, το βάρος και το ύψος του ατόμου

Β. Αν η ηλικία είναι μεγαλύτερη από 18 ετών τότε

1. Να υπολογίζει το ΔΜΣ
2. Να ελέγχει την τιμή του ΔΜΣ από τον παραπάνω πίνακα και να τυπώνει τον αντίστοιχο χαρακτηρισμό

Αλλιώς να τυπώνει κατάλληλο μήνυμα «Δεν υπολογίζεται το ΔΜΣ»

2) Μια αλυσίδα κινηματογράφων έχει δέκα αίθουσες. Τα ονόματα των αιθουσών καταχωρούνται σε μία λίστα ΟΝ και οι μέσες μηνιαίες εισπράξεις κάθε αίθουσας για ένα έτος καταχωρούνται λίστα Ε. Να γράψετε αλγόριθμο ο οποίος:

α. να διαβάζει τα ονόματα των αιθουσών

β. να διαβάζει τις μηνιαίες εισπράξεις των αιθουσών αυτού του έτους

γ. να υπολογίζει τη μέση μηνιαία τιμή των εισπράξεων για κάθε αίθουσα

δ. να βρίσκει και να εμφανίζει τη μικρότερη μέση μηνιαία τιμή

ε. να βρίσκει και να εμφανίζει το όνομα ή τα ονόματα των αιθουσών που έχουν την ανωτέρω μικρότερη μέση μηνιαία τιμή.

Παρατήρηση: Θεωρήστε ότι οι μηνιαίες εισπράξεις είναι θετικοί αριθμοί.

ΑΣΚΗΣΗ 1

```
H=input('Dose hlikia')
Y=input('Dose ypsos')
B=input('Dose varos')

if H>18:
   DMS=B/Y*2
   if DMS<18.5:
      print "Adinato"
   if DMS >= 18.5 and DMS< 25:
      print "kanoniko"
   if DMS >=25 and DMS<30:
      print "vari"
   if DMS >=30:
      print "Ypervaro"

else:
   print "Den ypologizete to DMS"
```

ΑΣΚΗΣΗ 2

```
ON=[]

E=[]

for x in range(10):
   S=0.0
   for y in range(12):
      print "Μηνας",y+1
e=input("Δώσε εισπράξεις")
S=S+e
   E.append(S/12)

min1=E[0]
for x in range(1,10,1):
   if E[x]< min1:
      min1=E[x]

print "μικρότερη μέση μηνιαία τιμή", min1

for x in range(10):
   if E[x]== min1:
      print ON[x]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2003

1) Κάποια δημοτική αρχή ακολουθεί την εξής τιμολογιακή πολιτική για την κατανάλωση νερού ανά μήνα:

Χρεώνει **πάγιο ποσό 2 ευρώ** και εφαρμόζει κλιμακωτή χρέωση σύμφωνα με τον παρακάτω πίνακα:

Κατανάλωση σε κυβικά μέτρα	**Χρέωση ανά κυβικό**
από 0 έως και 5	δωρεάν
από 5 έως και 10	0,5 ευρώ
από 10 έως και 20	0,7 ευρώ
από 20 και άνω	1,0 ευρώ

Στο ποσό που προκύπτει από την αξία του νερού και το πάγιο υπολογίζεται ο **Φ.Π.Α. με συντελεστή 18%**. Το τελικό ποσό προκύπτει από την άθροιση της αξίας του νερού, το πάγιο, το Φ.Π.Α. και το **δημοτικό φόρο που είναι 5 ευρώ**.

Να γράψετε αλγόριθμοσε Python ο οποίος:

α. Να διαβάζει τη μηνιαία κατανάλωση του νερού.

β. Να υπολογίζει την αξία του νερού που καταναλώθηκε σύμφωνα με την παραπάνω τιμολογιακή πολιτική.

γ. Να υπολογίζει το Φ.Π.Α.

δ. Να υπολογίζει και να εκτυπώνει το τελικό ποσό.

2)Κατά τη διάρκεια πρωταθλήματος μπάσκετ μια ομάδα πουαποτελείται από δώδεκα(12) παίκτες έδωσε είκοσι(20) αγώνες, στους οποίους συμμετείχαν όλοι οι παίκτες.

Να αναπτύξετε στο τετράδιό σας αλγόριθμο σε Pythono οποίος:

α. Να διαβάζει τα ονόματα των παικτών και να ταποθηκεύει στη λίστα ΟΝ.

β. Να διαβάζει τους πόντους που σημείωσε κάθε παίκτης σε κάθε αγώνα.

γ. Να υπολογίζει για κάθε παίκτη το συνολικό αριθμόπόντων του σε όλους τους αγώνες και το μέσο όροπόντων ανά αγώνα και να τα αποθηκεύει στις λίστες SUMκαι ΜΟ.

δ. Να εκτυπώνει τα ονόματα των παικτών της ομάδας καιτο μέσο όρο πόντων του κάθε παίκτη ταξινομημένα μεβάση το μέσο όρο τους κατά φθίνουσα σειρά.

Παρατήρηση: Σε περίπτωση ισοβαθμίας δεν μας ενδιαφέρει η σχετική σειρά των παικτών.

ΑΣΚΗΣΗ 1

```
K=input("Δωσε κατανάλωση νερού")

if K>=1 and K<=5:
   Xr=K*0+2
if K>=6 and K<=10:
   Xr=5*0+(K-5)*0.5+2
if K>=11 and K<=20:
   Xr=5*0+5*0.5+(K-10)*0.7+2
if K>20:
   Xr=5*0+5*0.5+10*0.7+(K-20)*1.0+2

FPA=Xr*18/100

TP=Xr+FPA+5

print "τελικό ποσό", TP
```

ΑΣΚΗΣΗ 2

```
ON=[]
SUM=[]
MO=[]

for x in range(12):
   on=str(raw_input("Δωσε ονομα"))
   S=0.0
   for y in rnage(20):
      p=input("Δωσε πόντους ")
      S=S+p
   ON.append(on)
   SUM.append(S)
   MO.append(S/20)

N=12
for i in range(1,N,1):
   for j in rnage(N-1,i-1,-1):
      if MO[j]>MO[j-1]:
         MO[j], MO[j-1]= MO[j-1], MO[j]
         SUM[j], SUM[j-1]= SUM[j-1], SUM[j]
         ON[j], ON[j-1]= ON[j-1], ON[j]

for x in range(20):
   print ON[x], MO[x]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣΕΣΠΕΡΙΝΟΥ ΓΕΛ 2003

1) Για κάθε υπάλληλο δίνονται: ο μηνιαίος βασικός μισθός και ο αριθμός των παιδιών του. Δεχόμαστε ότι ο υπάλληλος μπορεί να έχει μέχρι και20 παιδιά και ότι ο μηνιαίος βασικός μισθός του κυμαίνεται από 500 μέχρι και 1000 ευρώ.

Οι συνολικές αποδοχές του υπολογίζονται ως το άθροισμα του μηνιαίου βασικού μισθού και του οικογενειακού επιδόματός του. Το οικογενειακό επίδομα υπολογίζεται ως εξής:

30 ευρώ για κάθε παιδί μέχρι και τρία παιδιά, και**40 ευρώ** για κάθε παιδί πέραν των τριών(4ο, 5ο, 6ο κ.τ.λ.).

Να γράψετε αλγόριθμο σε Python, ο οποίος:

α. εισάγει τα κατάλληλα δεδομένα και ελέγχει την ορθή καταχώρισή τους,

β. υπολογίζει και εμφανίζει το οικογενειακό επίδομα και

γ. υπολογίζει και εμφανίζει τις συνολικές αποδοχές του υπαλλήλου.

2)Για κάθε μαθητή δίνονται τα στοιχεία: ονοματεπώνυμο, προφορικός και γραπτός βαθμός ενός μαθήματος.

Να γραφεί αλγόριθμος σε Python, ο οποίος εκτελεί τις ακόλουθεςλειτουργίες:

α. Διαβάζει τα στοιχεία πολλών μαθητών και σταματά όταν δοθεί ως ονοματεπώνυμο το κενό.

β. Ελέγχει αν ο προφορικός και ο γραπτός βαθμός είναιαπό0 μέχρι και20.

γ. Υπολογίζει τον τελικό βαθμό του μαθήματος, ο οποίοςείναι το άθροισμα του30% του προφορικού βαθμού καιτου70% του γραπτού βαθμού. Επίσης, τυπώνει τοονοματεπώνυμο του μαθητή και τον τελικό βαθμό του μαθήματος.

δ. Υπολογίζει και τυπώνει το ποσοστό των μαθητών πουέχουν τελικό βαθμό μεγαλύτερο του18.

ΑΣΚΗΣΗ 1

```
M=input("Δώσε μηνιαίο βασικό μισθό")
whileM<500 orM>1000:
M=input("Δώσε ΣΩΣΤΟ μηνιαίο βασικό μισθό")

P=input("Δώσε αριθμό παιδιών")
whileP>20:
P=input("Δώσε ΣΩΣΤΟ αριθμό παιδιών")


if P<=3:
   EP=P*30

if P>=4:
   EP=P*40

print "οικογενειακό επίδομα ",EP

print "συνολικές αποδοχές", M+EP
```

ΑΣΚΗΣΗ 2

```
c=0
c1=0

on=raw_input("Δώσε ονομα")

whileon!=" " :
c1=c1+1

p=input("Δώσε προφορικό βαθμό")
whilep<0 orp>20:
p=input("Δώσε ΣΩΣΤΟ προφορικό βαθμό")


g=input("Δώσε γραπτό βαθμό")
whileg<0 org>20:
g=input("Δώσε ΣΩΣΤΟ γραπτό βαθμό")

V=p*30/100 + g*70/100

   print on, V

   if  V>18:
      c=c+1
   on=raw_input("Δώσε ονομα")

print "ποσοστό  με τελικό βαθμό μεγαλύτερο του 18",100*c/c1
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2004

1) Μία εταιρεία ταχυδρομικών υπηρεσιών εφαρμόζει για τα έξοδα αποστολής ταχυδρομικών επιστολών εσωτερικού και εξωτερικού, χρέωση σύμφωνα με τον παρακάτω πίνακα

Βάρος επιστολής σε γραμμάρια	**Χρέωση εσωτερικού σε Ευρώ**	**Χρέωση εξωτερικού σε Ευρώ**
από 0 έως και 500	2,0	4,8
από 500 έως και 1000	3,5	7,2
από 1000 έως και 2000	4,6	11,5

Για παράδειγμα τα έξοδα αποστολής μιας επιστολής βάρους800 γραμμαρίων και προορισμού εσωτερικού είναι 3,5Ευρώ.

Να γράψετε αλγόριθμο σε Pythono οποίος:

α. Να διαβάζει το βάρος της επιστολής.

β. Να διαβάζει τον προορισμό της επιστολής. Η τιμή " ΕΣ" δηλώνει προορισμό εσωτερικού και η τιμή "ΕΞ" δηλώνει προορισμό εξωτερικού.

γ. Να υπολογίζει τα έξοδα αποστολής ανάλογα με τον προορισμό και το βάρος της επιστολής.

δ. Να εκτυπώνει τα έξοδα αποστολής.

Παρατήρηση. Θεωρείστε ότι ο αλγόριθμος δέχεται τιμές γιατο βάρος μεταξύ του 0 και του 2000 και για τον προορισμό μόνο τις τιμές "ΕΣ" και "ΕΞ".

2)Για την πρώτη φάση της Ολυμπιάδας Πληροφορικής δήλωσαν συμμετοχή 500 μαθητές. Οι μαθητές διαγωνίζονται σε τρεις γραπτές εξετάσεις και βαθμολογούνται με ακέραιους βαθμούς στη βαθμολογική κλίμακα από 0 έως και 100.

Να γράψετε αλγόριθμο σε Pythono οποίος:

α. Να διαβάζει τα ονόματα των μαθητών και να τα αποθηκεύει σε λίστα ΟΝ.

β. Να διαβάζει τους τρεις βαθμούς που έλαβε κάθε μαθητής και να τους αποθηκεύει στιςλίστες Α, Β, C.

γ. Να υπολογίζει το μέσο όρο των βαθμών του κάθε μαθητήκαι να τον αποθηκεύει στην λίστα Μ.

δ. Να εκτυπώνει τα ονόματα των μαθητών και δίπλα τους το μέσο όρο των βαθμών τους ταξινομημένα με βάση τον μέσο όρο κατά φθίνουσα σειρά. Σε περίπτωση ισοβαθμίας η σειρά ταξινόμησης των ονομάτων να είναι αλφαβητική.

ε. Να υπολογίζει και να εκτυπώνει το πλήθος των μαθητών με το μεγαλύτερο μέσο όρο.

Παρατήρηση: Θεωρείστε ότι οι βαθμοί των μαθητών είναι μεταξύτου 0 και του 100

ΑΣΚΗΣΗ 1

```
B=input("Δώσε βάρος επιστολής")
while B<0 or B>2000:
   B=input("Δώσε ΣΩΣΤΟ βάρος επιστολής")

P=raw_input("Δωσε προορισμό επιστολής")
while P not in ["ΕΣ", "ΕΞ"]:
P=raw_input("Δωσε ΣΩΣΤΟ προορισμό επιστολής")

if P=="ΕΣ":
   if B>=0 and B<=500:
      A=2.0
   if B>=501 and B<=1000:
      A=3.5
   if B>=1001 and B<=2000:
      A=4.6

if P=="ΕΞ":
   if B>=0 and B<=500:
      A=4.8
   if B>=501 and B<=1000:
      A=7.2
   if B>=1001 and B<=2000:
A=11.5

print "έξοδα αποστολής:", A
```

ΑΣΚΗΣΗ 2

```
ON=[] ; A=[] ; B=[] ; C=[] ; M=[]

for x in range(500):
   name=str(raw_input("Δωσεονομα"))
   ON.appens(name)

   vathmos1=input("Δωσεβαθμό Α")
   while vatmos1<1 or vathmos1>100:
        vathmos1=input("ΔωσεΣΩΣΤΟβαθμό")
   A.append(vathmos1)

   vathmos2=input("Δωσεβαθμό Β")
   while vatmos2<1 or vathmos2>100:
        vathmos2=input("ΔωσεΣΩΣΤΟβαθμό")
   B.append(vathmos2)

   vathmos3=input("Δωσεβαθμό C")
   while vatmos3<1 or vathmos3>100:
        vathmos3=input("ΔωσεΣΩΣΤΟβαθμό")
   C.append(vathmos3)

for x in range(500):
   MO=(A[x]+B[x]+C[x])/3.0
   M.append(MO)

N=len(M)

for i in range(1,N,1):
   for j in range(N-1,i-1,-1):

      if   M[j] > M[j-1]:
         M[j],M[j-1]=M[j-1],M[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]
         C[j],C[j-1]=C[j-1],C[j]

      if   M[j]==M[j-1]   and   ON[j]<ON[j-1]:
         M[j],M[j-1]=M[j-1],M[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]
         C[j],C[j-1]=C[j-1],C[j]

for x in range(500):
   print ON[x], M[x]

c=0
for x in range(500):
if M[x]==M[0]:
      c=c+1
print "πλήθος μαθητών με το μεγαλύτερο μέσο όρο", c
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2004

1) Σε κάποια εξεταστική δοκιμασία κάθε γραπτό αξιολογείται αρχικά από δύο βαθμολογητές και υπάρχει περίπτωση το γραπτό να χρειάζεται αναβαθμολόγηση από τρίτο βαθμολογητή όταν υπάρχει διαφορά βαθμολογίας ανάμεσα στον $1^ο$ και $2^ο$ βαθμολογητή μεγαλύτερη από 15 μονάδες. Στην περίπτωση αναβαθμολόγησης ο τελικός βαθμός υπολογίζεται ως εξής:

i. Αν ο βαθμός του τρίτου βαθμολογητή είναι ίσος με το μέσο όρο (Μ.Ο.) των βαθμών των δύο πρώτων βαθμολογητών, τότε ο τελικός βαθμός είναι ο Μ.Ο.

ii. Αν ο βαθμός του τρίτου βαθμολογητή είναι μικρότερος από το μικρότερο βαθμό (MIN) των δύο πρώτων βαθμολογητών, τότε ο τελικός βαθμός είναι ο MIN.

iii. Διαφορετικά, ο τελικός βαθμός είναι ο μέσος όρος του βαθμού του τρίτου βαθμολογητή με τον πλησιέστερο προς αυτόν βαθμό των δύο πρώτων βαθμολογητών.

Να αναπτύξετε αλγόριθμο σε Python υπολογισμού του τελικού βαθμού ενός γραπτού με αναβαθμολόγηση, ο οποίος:

α. να διαβάζει τους βαθμούς του πρώτου, του δεύτερου και του τρίτου βαθμολογητή ενός γραπτού.

β. να υπολογίζει και να εκτυπώνει το μεγαλύτερο (MAX) και το μικρότερο (MIN) από τους βαθμούς του πρώτου και του δεύτερου βαθμολογητή.

γ. να υπολογίζει και να εκτυπώνει τον τελικό βαθμό του γραπτού σύμφωνα με την παραπάνω διαδικασία.

Παρατήρηση: Θεωρήστε ότι και οι τρεις βαθμοί είναι θετικοί ακέραιοι αριθμοί μεταξύ 1 και 100 μονάδες και δεν απαιτείται έλεγχος των δεδομένων.

2) Σε κάποια χώρα της Ευρωπαϊκής Ένωσης διεξάγονται εκλογές για την ανάδειξη των μελών του Ευρωπαϊκού Κοινοβουλίου. Θεωρήστε ότι μετέχουν15 συνδυασμοί κομμάτων, οι οποίοι θα μοιραστούν 24 έδρες σύμφωνα με το ποσοστό των έγκυρων ψηφοδελτίων που έλαβαν. Κόμματα που δεν συγκεντρώνουν ποσοστό έγκυρων ψηφοδελτίων τουλάχιστον ίσο με το 3% του συνόλου των έγκυρων ψηφοδελτίων δεν δικαιούνται έδρα.

Για κάθε κόμμα, εκτός του πρώτου κόμματος, ο αριθμός των εδρών που θα λάβει υπολογίζεται ως εξής: Το ποσοστό των έγκυρων ψηφοδελτίων πολλαπλασιάζεται επί 24 και στη συνέχεια το γινόμενο διαιρείται με το άθροισμα των ποσοστών όλων των κομμάτων που δικαιούνται έδρα. Το ακέραιο μέρος του αριθμού που προκύπτει είναι ο αριθμός των εδρών που θα λάβει το κόμμα.

Το πρώτο κόμμα λαμβάνει τις υπόλοιπες έδρες.

Να γράψετε αλγόριθμο σε Pythono οποίος:

α. να διαβάζει και να αποθηκεύει σε λίστες τα ονόματα των κομμάτων και τα αντίστοιχα ποσοστά των έγκυρων ψηφοδελτίων τους.

β. να εκτυπώνει τα ονόματα και το αντίστοιχο ποσοστό έγκυρων ψηφοδελτίων των κομμάτων που δεν έλαβαν έδρα.

γ. να εκτυπώνει το όνομα του κόμματος με το μεγαλύτερο ποσοστό έγκυρων ψηφοδελτίων.

δ. να υπολογίζει και να εκτυπώνει το άθροισμα των ποσοστών όλων των κομμάτων που δικαιούνται έδρα.

ε. να εκτυπώνει τα ονόματα των κομμάτων που έλαβαν έδρα και τον αντίστοιχο αριθμό των εδρών τους.

Παρατηρήσεις:

α) Υποθέτουμε ότι δεν υπάρχουν δύοκόμματα που να έχουν το ίδιο ποσοστόέγκυρων ψηφοδελτίων.

β) Μπορείτε να χρησιμοποιήσετε τησυνάρτηση int() που επιστρέφει τοακέραιο μέρος του πραγματικού αριθμούx.

γ) Τα ποσοστά να θεωρηθούν επί τοις εκατό(%).

ΑΣΚΗΣΗ 1

```
V1=input("Δωσε Βαθμό Α")

V2=input("Δωσε Βαθμό Β")

V3=input("Δωσε Βαθμό Γ")

MO=(V1+V2)/2.0

if V1>V2:
   MIN=V2
   MAX=V1
if V1<V2:
   MIN=V1
   MAX=V2

print "μεγαλύτερος βαθμός ", MAX
print "μικρότερος βαθμός ", MIN

if  abs(V1-V2) > 15 :

   if V3==MO:
      TV=MO

   elif V3<MIN :
      TV=MIN

   else:
      if abs(V3-V1) > abs(V3-V2):
         TV=(V3+V2)/2.0
      if abs(V3-V1) < abs(V3-V2):
         TV=(V3+V1)/2.0

print "τελικός  βαθμός ", TV
```

ΑΣΚΗΣΗ 2

```
ON=[]
P=[]

for x in range(15):

   on=str(raw_input("Δωσε ονομα"))
   p=input("Δωσε ποσοστό")
   On.append(on)
   P.append(p)


for x in range(15):
   if P[x]<3:
      print "δεν  έλαβαν έδρα",ON[x], P[x]

max1=0

for x in range(15):
   if P[x]>max1:
      max1=P[x]
      Y=x

print "μεγαλύτερο ποσοστό ", On[Y]

S=0
for x in range(15):
   if P[x]>=3:
      S=S+P[x]

      print "έλαβαν έδρα",ON[x], P[x]

print "άθροισμα  ποσοστών κομμάτων που δικαιούνται έδρα", S
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣΕΣΠΕΡΙΝΟΥ ΓΕΛ 2004

1) Σε έναν αγώνα δισκοβολίας συμμετέχουν 20 αθλητές. Κάθε αθλητής έκανε μόνο μία έγκυρη ρίψη που καταχωρείται ως επίδοση του αθλητή και εκφράζεται σε μέτρα.

Να αναπτύξετε αλγόριθμο σε Python που

α. να διαβάζει για κάθε αθλητή το όνομα και την επίδοσή του,

β. να ταξινομεί τους αθλητές ως προς την επίδοσή τους,

γ. να εμφανίζει τα ονόματα και τις επιδόσεις των τριών πρώτων αθλητών, αρχίζοντας από εκείνον με την καλύτερη επίδοση,

δ. να εμφανίζει τα ονόματα και τις επιδόσεις των πέντε τελευταίων αθλητών, αρχίζοντας από εκείνον με την καλύτερη επίδοση.

Σημείωση: Να θεωρήσετε ότι δεν υπάρχουν αθλητές με την ίδια ακριβώς επίδοση.

2) Μία εταιρεία απασχολεί 30 υπαλλήλους. Οι μηνιαίες αποδοχές κάθε υπαλλήλου κυμαίνονται από 0€ έως και 3.000€.

Α. Να γράψετε αλγόριθμο σε Python που για κάθε υπάλληλο

1. να διαβάζει το ονοματεπώνυμο και τις μηνιαίες αποδοχές και να ελέγχει την ορθότητα καταχώρησης των μηνιαίων αποδοχών του,

2. να υπολογίζει το ποσό του φόρου κλιμακωτά,σύμφωνα με τον παρακάτω πίνακα:

Μηνιαίες αποδοχές	**Ποσοστό κράτησης φόρου**
Έως και700 €	0%
Άνω των700 € έως και1.000 €	15%
Άνω των1.000 € έως και1.700 €	30%
Άνω των1.700 €	40%

3. να εμφανίζει το ονοματεπώνυμο, τις μηνιαίες αποδοχές, το φόρο και τις καθαρές μηνιαίες αποδοχές, που προκύπτουν μετά την αφαίρεση του φόρου.

Β. Τέλος, ο παραπάνω αλγόριθμος να υπολογίζει και να εμφανίζει

1. το συνολικό ποσό που αντιστοιχεί στο φόρο όλων των υπαλλήλων,

2. το συνολικό ποσό που αντιστοιχεί στις καθαρές μηνιαίες αποδοχές όλων των υπαλλήλων.

ΑΣΚΗΣΗ 1

```
ON=[]
E=[]

for x in range (20):
   on=raw_input("Δώσε ονομα")
   e=input("Δώσε επίδοση")

   ON.append(on)
   E.append(e)

N=20
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if E[j]>E[j-1]:
         E[j],E[j-1]=E[j-1],E[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]


print "Τρείς πρώτοι", ON[0],ON[1],On[2]

print "Πέντε τελευταίοι", ON[-5],ON[-4],ON[-3], ON[-2],ON[-1]
```

ΑΣΚΗΣΗ 2

```
SF=0
SE=0

for x in range(30):
   on=raw_input("Δώσε ονομα")
   P=input("Δώσε μηνιαίες αποδοχές ")
   while P<0 orP>3000:
        P=input("Δώσε ΣΩΣΤΕΣ μηνιαίες αποδοχές ")

   if P>=0 and P<=700:
      F=P*0/100
   if P>=701 and P<=1000:
      F=0 * 700/100+(P-700)*15/100
   if P>=1001 and P<=1700:
      PRO=0 * 700/100+300*15/100 + (P-1000)*30/100
   if P>=1701 :
      PRO=0 * 700/100+300*15/100 + 700*30/100 + (P-1700)*40/100


   SF=SF+F

   SE=SE+P-F

   print on, P, F, P-F

print "συνολικός  φόρος", SF

print "συνολικές καθαρές μηνιαίες αποδοχές ", SE
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2014

1)Μια εταιρεία δημοσκοπήσεων θέτει σ' ένα δείγμα2000πολιτών ένα ερώτημα. Για την επεξεργασία των δεδομένωννα αναπτύξετε αλγόριθμο σε Python που:

Α. να διαβάζει το φύλο του πολίτη(Α=Άνδρας, Γ=Γυναίκα)και να ελέγχει την ορθή εισαγωγή

Β. να διαβάζει την απάντηση στο ερώτημα, η οποία μπορεί να είναι « ΝΑΙ», «ΟΧΙ», «ΔΕΝ ΞΕΡΩ» και να ελέγχει την ορθή εισαγωγή

Γ. να υπολογίζει και να εμφανίζει το πλήθος των ατόμων που απάντησαν «ΝΑΙ»

Δ. στο σύνολο των ατόμων που απάντησαν «ΝΑΙ» να υπολογίζει και να εμφανίζει το ποσοστό των ανδρών και το ποσοστό των γυναικών.

2) Σ ' ένα διαγωνισμό συμμετέχουν 5000 διαγωνιζόμενοι και εξετάζονται σε δύο μαθήματα.

Να γράψετε αλγόριθμο σε Python που:

Α. να διαβάζει και να καταχωρίζει σε κατάλληλες λίστες για κάθε διαγωνιζόμενο τον αριθμό μητρώου, το ονοματεπώνυμο και τους βαθμούς που πήρε στα δύο μαθήματα.

Οι αριθμοί μητρώου θεωρούνται μοναδικοί. Η βαθμολογική κλίμακα είναι από 0 έως και 100.

Β. να εμφανίζει κατάσταση επιτυχόντων με την εξής μορφή:

Αριθ. Μητρώου Ονοματεπώνυμο Μέσος Όρος

Επιτυχών θεωρείται ότι είναι αυτός που έχει μέσο όρο βαθμολογίας μεγαλύτερο ή ίσο του 60.

Γ. να διαβάζει έναν αριθμό μητρώου και

α. σε περίπτωση που ο αριθμός μητρώου είναι καταχωρισμένος στη λίστα, να εμφανίζεται ο αριθμός μητρώου, το ονοματεπώνυμο, ο μέσος όρος βαθμολογίας και η ένδειξη «ΕΠΙΤΥΧΩΝ» ή «ΑΠΟΤΥΧΩΝ», ανάλογα με τον μέσο όρο.

β. σε περίπτωση που ο αριθμός μητρώου δεν είναι καταχωρισμένος στη λίστα, να εμφανίζεται το μήνυμα «Ο αριθμός μητρώου δεν αντιστοιχεί σε διαγωνιζόμενο».

Σημείωση: Δεν απαιτείται έλεγχος εγκυρότητας καταχώρισης δεδομένων για τον αριθμό μητρώου.

ΑΣΚΗΣΗ 1

```
for x in range(2000):
   F=raw_input("Δωσε το φύλο ")
   while F not in ["Α", "Γ"]:
      F=raw_input("Δωσε ΣΩΣΤΟ  φύλο ")

    E=raw_input("Δωσε απάντηση στο ερώτημα")
    while E not in ["ΝΑΙ", "ΟΧΙ", "ΔΕΝ ΞΕΡΩ"]:
      E=raw_input("Δωσε ΣΩΣΤΗ απάντηση στο ερώτημα")

   S=0
   SA=0
   SG=0
   if E=="ΝΑΙ":
      S=S+1
      if F=="Α":
         SA=SA+1
      if  F=="Γ":
         SG=SG+1
print "πλήθος ατόμων που απάντησαν ΝΑΙ",S

print "ποσοστό των ανδρών που απάντησαν ΝΑΙ",SA*100/S

print "ποσοστό των γυναικών που απάντησαν ΝΑΙ",SG*100/S
```

ΑΣΚΗΣΗ 2

```
ON=[]
V1=[]
V2=[]
AM=[]

for x in range(5000):
   on=str(raw_input("Δωσε ονομα"))
   ON.append(on)

   am=input("Δώσε αριθμό μητρώου")
   AM.append(am)

   v1=input("Δώσε βαθμό 1")
   while v1<0 or v1>100:
      v1=input("Δώσε ΣΩΣΤΟ βαθμό 1")
   V1.append(v1)

   v2=input("Δώσε βαθμό 2")
   while v2<0 or v2>100:
      v2=input("Δώσε ΣΩΣΤΟ βαθμό 2")
V2.append(v2)

for x in range(5000):
   if (V1[x]+V2[x])/2.0 >= 60:
      print AM[x], ON[x], (V1[x]+V2[x])/2.0

F=False
am=input("Δώσε αριθμό μητρώου")
for x in range(5000):
   if AM[x]==am:
      if  (V1[x]+V2[x])/2.0 > 60 :
          print AM[x], ON[x], (V1[x]+V2[x])/2.0, "ΕΠΙΤΥΧΩΝ"
     else:
          print AM[x], ON[x], (V1[x]+V2[x])/2.0, "ΑΠΟΤΥΧΩΝ"
      F=True
if F==False:
   print "Ο αριθμός μητρώου δεν αντιστοιχεί σε διαγωνιζόμενο"
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2005

1) Δίνεται λίστα Α[Ν] ακέραιων και θετικών αριθμών, καθώς και λίστα Β[Ν-1] πραγματικών και θετικών αριθμών.

Να γραφεί αλγόριθμος σε Python, ο οποίος να ελέγχει αν κάθε στοιχείο Β[i] είναι ο μέσος όρος των στοιχείων Α[i] και Α[i+1], δηλαδή αν Β[i] = (Α[i] + Α[i+1])/2.

Σε περίπτωση που ισχύει, τότε να εμφανίζεται το μήνυμα «Η λίστα Β είναι ο τρέχων μέσος του Α», διαφορετικά να εμφανίζεται το μήνυμα «Η λίστα Β δεν είναι ο τρέχων μέσος του Α».

Για παράδειγμα:

Έστω ότι τα στοιχεία της λίστας Α είναι: 1, 3, 5, 10, 15

και ότι τα στοιχεία της λίστας Β είναι: 2, 4, 7.5, 12.5.

Τότε ο αλγόριθμος θα εμφανίσει το μήνυμα «Η λίστα Β είναι ο τρέχων μέσος του Α», διότι 2 = (1+3)/2, 4=(3+5)/2, 7.5= (5+10)/2, 12.5=(10+15)/2.

2) Σ' ένα διαγωνισμό συμμετέχουν 100 υποψήφιοι. Κάθε υποψήφιος διαγωνίζεται σε 50 ερωτήσεις πολλαπλής επιλογής.

Να αναπτύξετε αλγόριθμο που να κάνει τα παρακάτω:

α. Να καταχωρεί σε λίστα Α τα αποτελέσματατων απαντήσεων του κάθε υποψηφίου σε κάθε ερώτηση.

Κάθε καταχώρηση μπορεί να είναι μόνο μία από τις παρακάτω:

i. **Σ** αν είναι σωστή η απάντηση

ii. **Λ** αν είναι λανθασμένη η απάντηση και

iii. **Ξ** αν ο υποψήφιος δεν απάντησε.

Να γίνεται έλεγχος των δεδομένων εισόδου.

β. Να βρίσκει και να τυπώνει τον αριθμότηςερώτησης που παρουσιάζει το μεγαλύτερο βαθμόδυσκολίας, δηλαδή έχει το μικρότερο πλήθος σωστώναπαντήσεων.

γ. Αν κάθε Σ βαθμολογείται με 2 μονάδες, κάθε Λ με -1μονάδα και κάθε Ξ με 0 μονάδες τότε

i. Να δημιουργεί μία λίστα Β, κάθε στοιχείο του οποίου θα περιέχει αντίστοιχα τησυνολική βαθμολογία ενός υποψηφίου.

ii. Να τυπώνει το πλήθος των υποψηφίων πουσυγκέντρωσαν βαθμολογία μεγαλύτερη από50.

ΑΣΚΗΣΗ 1

```
F=True

for i in range(len(A)-1):
   if (A[i]+ A[i+1])/2 != B[i]:
      F=False
ifF==False:
print "Η λίστα Β δεν είναι ο τρέχων μέσος του Α"
else:
print "Η λίστα Β είναι ο τρέχων μέσος του Α"
```

ΑΣΚΗΣΗ 2

```
A=[]  ;  B=[]

for x in range (100):
   T=[]
   for y in range(50):
      print "Ερωτηση:", y+1
      v=str(raw_input("Δωσε βαθμόλογία"))
      while v not in ["Σ", "Λ", "Ξ"]:
         v=str(raw_input("Δωσε Σ ή Λ ή Ξ"))
T.append(v)
   A.append(T)

min1=101
for x in range (50):
   c=0
   for y in range(100):
      if A [y] [x] =="Λ" or A [y] [x] =="Ξ":
         c=c+1
   if c<min1:
      min1=c
      Z=x

print "αριθμό ερώτησης με μεγαλύτερο βαθμό δυσκολίας", Z

for x in range (100):
   S=0
   for y in range(50):
      if A [x] [y] =="Σ":
         S=S+2
      if A [x] [y] =="Λ":
         S=S-1
   B.append(S)

c=0
for x in B:
   ifx>50:
c=c+1
```

```
print "βαθμολογία μεγαλύτερη από 50", c
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2005

1)Εκατό(100) υποψήφιοι του ΑΣΕΠ διαγωνίζονται σε τρίαμαθήματα για την κάλυψη θέσεων του Δημοσίου.

Να γραφείκύριο πρόγραμμα σε Python που να κάνει τα παρακάτω:

α) Διαβάζει τα ονόματα των100 υποψηφίων του ΑΣΕΠ και τη βαθμολογία καθενός υποψηφίου σε τρία διαφορετικάμαθήματα.

(Θεωρήστε ότι η βαθμολογία κάθε μαθήματος είναι από 1 έως 20).

β) Βρίσκει και τυπώνει τον ελάχιστο και τον μέγιστο βαθμόκαθενός υποψηφίου στα τρία μαθήματα που εξετάστηκε.

γ) Να γραφεί υποπρόγραμμα, το οποίο να καλείται από τοκύριο πρόγραμμα, για τον υπολογισμό και την εκτύπωσητου μέσου όρου κάθε υποψηφίου στα τρία μαθήματα πουδιαγωνίστηκε.

2) Μια αεροπορική εταιρία ταξιδεύει σε 15 προορισμούς του εσωτερικού. Στα πλαίσια της οικονομικής πολιτικής πουπρόκειται να εφαρμόσει, κατέγραψε το ποσοστό πληρότηταςτων πτήσεων για κάθε μήνα του προηγούμενου ημερολογιακού έτους.

Η πολιτική έχει ως εξής:

- Δεν θα γίνει καμία περικοπή σε προορισμούς, στους οποίους το μέσο ετήσιο ποσοστό πληρότητας των πτήσεωνείναι μεγαλύτερο του65.

- Θα γίνουν περικοπές πτήσεων σε προορισμούς, στους οποίους το μέσο ετήσιο ποσοστό πληρότητας των πτήσεωνκυμαίνεται από40 έως και65.

- Θα καταργηθούν οι προορισμοί, στους οποίους το μέσοετήσιο ποσοστό πληρότητας των πτήσεων είναι μικρότεροτου40.

Να γραφεί αλγόριθμος σε Python ο οποίος:

1. Να διαβάζει τα ονόματα των15 προορισμών και να ταααποθηκεύει σε μια λίστα ΟΝ.

2. Να διαβάζει τα ποσοστά πληρότητας των πτήσεων των15 προορισμών για κάθε μήνα και να υπολογίζει και να αποθηκεύει σε λίστα Ρτο μέσοετήσιο ποσοστό πληρότητας. Οι τιμές κάθε μήνα πρέπει να είναι μεταξύ 1 και 100.

3. Να βρίσκει και να τυπώνει τα ονόματα των προορισμώνπου δεν θα γίνει καμία περικοπή πτήσεων.

4. Να βρίσκει και να τυπώνει τα ονόματα των προορισμώνπου θα καταργηθούν.

5. Να βρίσκει και να τυπώνει τα ονόματα τωνπροορισμών, στους οποίους θα γίνουν περικοπές πτήσεων.

ΑΣΚΗΣΗ 1

```
def MESOS_OROS(V1,V2,V3,ON):
   for x in range(100):
      print "Ο υποψήφιος ", ON[x], "έχει Μέσο Όρο", (V1[x]+V2[x]+V3[x])/3.0

ON=[]
V1=[]
V2=[]
V3=[]

for x in range(100):
   on=str(raw_input("Δωσε ονομα"))
   v1=input("Δωσε βαθμό στο μαθημα 1")
v2=input("Δωσε βαθμό στο μαθημα 2")
v3=input("Δωσε βαθμό στο μαθημα 3")

On.append(on)
   V1.append(v1)
   V2.append(v2)
   V3.append(v3)

for x in range(100):
   max1=V1[x]

   if V2[x]>max1:
      max1=V2[x]
   if V3[x]>max1:
      max1=V3[x]

   print "Ο υποψήφιος ", ON[x], "έχει μέγαλύτερο βαθμό ",max1

min1=V1[x]

   if V2[x]>min1:
      min1=V2[x]
   if V3[x]>min1:
      min1=V3[x]

print "Ο υποψήφιος ", ON[x], "έχει μικρότερο βαθμό ",min1

MESOS_OROS(V1,V2,V3,ON)
```

ΑΣΚΗΣΗ 2

```
ON=[]
P=[]

for x in range(15):
   on.str(raw_input("Δώσε ονομα προορισμού"))
ON.append(on)
   S=0
   for y in rage(12):
      printy+1, "Μήνας"
p=input("Δωσε ποσοστά πληρότητας")
whilep<1 orp>100:
p=input("Δωσε ΣΩΣΤΑ ποσοστά πληρότητας")
S=S+p

   EP=S/12

   P.append(EP)

for x in range(15):
   if P[x]>65:
      print "δεν θα γίνει καμία περικοπή", ON[x]

if P[x]<40:
      print "θα καταργηθεί", ON[x]

   if P[x]>=40 and P[x]<=65:
      print "θα  γίνουν  περικοπές", ON[x]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣΕΣΠΕΡΙΝΟΥ ΓΕΛ 2005

1) Για την εύρεση πόρων προκειμένου οι μαθητές της Δ΄ τάξης Εσπερινού Λυκείου να συμμετάσχουν σε εκδρομή οργανώνεται λαχειοφόρος αγορά. Οι μαθητές του Λυκείου διαθέτουν λαχνούς στα σχολεία της περιοχής τους. Διακόσιοι μαθητές από δεκαπέντε διαφορετικά σχολεία αγόρασαν ο καθένας από έναν μόνο λαχνό. Μετά από κλήρωση ένας μαθητής κερδίζει τον πρώτο λαχνό. Να γίνει τμήμα αλγορίθμου σε Pythonπου

α) για κάθε μαθητή που αγόρασε λαχνό να εισάγει σε λίστα Α 200 θέσεων το επώνυμό του και στην αντίστοιχη θέση λίστας Β 200 θέσεων το όνομα του σχολείου του,

β) να εισάγει σε λίστα Σ 15 θέσεων τα ονόματα όλων των σχολείων της περιοχής και στις αντίστοιχες θέσεις λίσταςΜ 15 θέσεων τις ηλεκτρονικές διευθύνσεις των σχολείων,

γ) να διαβάζει το επώνυμο του μαθητή, που κέρδισε τον πρώτο λαχνό,

δ) χρησιμοποιώντας τον αλγόριθμο της σειριακής αναζήτησης να προσδιορίζει τη θέση του επωνύμου του τυχερού μαθητή στην λίστα Α. Στη συνέχεια στην λίστα Β να βρίσκει το όνομα του σχολείου που φοιτά,

ε) λαμβάνοντας υπόψη το όνομα του σχολείου που φοιτά ο τυχερός μαθητής και χρησιμοποιώντας τον αλγόριθμο της σειριακής αναζήτησης να προσδιορίζει την θέση του σχολείου στην λίστα Σ. Στη συνέχεια στην λίστα Μ να βρίσκει τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του σχολείου αυτού,

στ) να εμφανίζει το επώνυμο του τυχερού μαθητή, το όνομα του σχολείου του και τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του σχολείου του.

Σημείωση: Να θεωρήσετε ότι δεν υπάρχουν μαθητές με το ίδιο επώνυμο και ότι κάθε μαθητής αγόρασε έναν μόνο λαχνό.

2) Σε ένα πανελλήνιο σχολικό διαγωνισμό μετέχουν 20 σχολεία. Κάθε σχολείο αξιολογεί 5 άλλα σχολεία και δεν αυτοαξιολογείται. Η βαθμολογία κυμαίνεται από 1 έως και 10.

Να γραφεί τμήμα αλγορίθμου σε Python που

α) να διαβάζει τα ονόματα των σχολείων και να τα αποθηκεύει σε λίστα Α20 θέσεων,

β) Να καταχωρίζει στην λίστα Β τη βαθμολογία που δίνει κάθε σχολείο για 5 άλλα σχολεία.

Σημείωση: Η λίστα Β θα περιέχει τις βαθμολογίες κάθε σχολείου με την μορφή υπολίστας

γ) να υπολογίζει τη συνολική βαθμολογία του κάθε σχολείου και να την καταχωρίζει σε λίστα 20 θέσεων με όνομα SUM,

δ) να εμφανίζει τα ονόματα και τη συνολική βαθμολογία όλων των σχολείων κατά φθίνουσα σειρά της συνολικής βαθμολογίας.

ΑΣΚΗΣΗ 1

```
def    binarysearch(A, eponimo) :
   first = 0
   last = len(A)-1
   found = False
   while   found== False  and  first <= last  :

      mid = ( first + last ) // 2
      if    A[ mid ] == eponimo :
         found = True
      elif   A[ mid ] < eponimo :
         first = mid + 1
      else:
         last = mid-1

   if   found == True:
        return   mid


A=[]
B=[]
S=[]
M=[]

for x in range(200):
   on=str(raw_input("Δώσε επώνυμό μαθητή"))
   onS=str(raw_input("Δώσε όνομα σχολείου "))

A.append(on)
   B.append(onS)

for x in range(15):
   onS=str(raw_input("Δώσε όνομα σχολείου"))
   em=str(raw_input("Δώσε ηλεκτρονική διευθύνση"))

S.append(on)
   M.append(em)

eponimo=raw_input("Δώσε επώνυμο μαθητή που κέρδισε")

SXOLEIO=B[binarysearch(A, eponimo)]

email=M[binarysearch(S, SXOLEIO)]

print eponimo, SXOLEIO, email
```

ΑΣΚΗΣΗ 2

```
def  bubbleSort(A,B,C):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]>A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
            C[j],C[j-1]=C[j-1],C[j]

A=[]
B=[]
SUM=[]

for x in range(20):
   on=str(raw_input("Δώσε ονομα σχολείου"))
A.append(on)
   T=[]
   for y in range(5):
      print "Αξιολόγηση:", y+1
      a=input("Δώσε αξιολόγηση")
      T.append(a)
   B.append(T)


for x in range(20):
   S=0
   for y in range(5):
      S=S+B[x][y]

   SUM.append(S)

bubbleSort(SUM,A,B)

for x in range(20):
   print A[x], SUM[x]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2005

1)Μία εμπορική εταιρεία μέσω αντιπροσώπων διαθέτει στοαγοραστικό κοινό τρεις τύπους προϊόντων Χ, Ψ και Ζ καιχορηγεί προμήθεια στους αντιπροσώπους της.

Να αναπτύξετε αλγόριθμο σε Python, ώστε

α) να διαβάζει τον τύπο ενός προϊόντος και την τιμή πώλησης αυτού,

β) να υπολογίζει κλιμακωτά την προμήθεια που θα δοθεί από την πώληση σύμφωνα με τον παρακάτω πίνακα:

Τιμή πώλησησε €	**Ποσοστάπρομήθειας**		
	Προϊόν Χ	**Προϊόν Ψ**	**Προϊόν Ζ**
Από 0 έως και 5.000	0%	2%	4%
Πάνω από 5.000 έως και 10.000	5%	6%	6%
Πάνω από 10.000	10%	7%	8%

Η είσοδος των δεδομένων και ο υπολογισμός της προμήθειας θα επαναλαμβάνεται μέχρι να δοθεί τύπος προϊόντος Τ,

γ) στο τέλος να εμφανίζεται

i. η προμήθεια που θα δοθεί για κάθε τύπο προϊόντος,

ii. η συνολική προμήθεια που έλαβαν οι αντιπρόσωποι.

2) Να γραφεί πρόγραμμα σε Python το οποίο:

Α. Να διαβάζει το πλήθος των ασθενών ενός νοσοκομείου, το οποίο δεν μπορεί να δεχτεί περισσότερους από 500 ασθενείς.

Β. Για κάθε ασθενή να διαβάζει τις ημέρες νοσηλείας του, τον κωδικό του ασφαλιστικού του ταμείου και τη θέση νοσηλείας. Να ελέγχει την ορθότητα εισαγωγής των δεδομένων συμφωνά με τα παρακάτω:

οι ημέρες νοσηλείας είναι αριθμός μεγαλύτερος ή ίσος του 1

τα ασφαλιστικά ταμεία είναι 10 με κωδικούς από 0 μέχρι και 9

οι θέσεις νοσηλείας είναι Α ή Β ή C

Γ. Να υπολογίζει και να εμφανίζει το μέσο όρο ημερών νοσηλείας των ασθενών στο νοσοκομείο.

Δ. Να υπολογίζει και να εμφανίζει για κάθε ασθενή το κόστος παραμονής που πρέπει να καταβάλει στο νοσοκομείο το ασφαλιστικό του ταμείο σύμφωνα με τις ημέρες και τη θέση νοσηλείας. Το κόστος θα υπολογιστεί με βάση τον παρακάτω πίνακα:

Θέση Νοσηλείας	**Κόστος παραμονής ανά ημέρα νοσηλείας**
Α	125 €
Β	90 €
Γ	60 €

Ε. Να υπολογίζει και να εμφανίζει με τη χρήση λίστας το συνολικό κόστος που θα καταβάλει το κάθε ασφαλιστικό ταμείο στο νοσοκομείο.

Ζ. Να υπολογίζει και να εμφανίζει το συνολικό ποσό που οφείλουν όλα τα ασφαλιστικά ταμεία στο νοσοκομείο.

Η. Να εμφανίζει τα ονόματα των τριών ταμείων που οφείλουν τα περισσότερα χρήματα στο Δημόσιο Σύστημα Υγείας.

ΑΣΚΗΣΗ 1

```
SP=0
SX=0
SY=0
SZ=0

TY=raw_input("Δώσε τον τύπο ενός προϊόντος")
whileTY!="T":

P=input("Δώσε τιμή προϊόντος")

if TY=="X" :
      if P>=1 and P<=5000:
         PRO=P*0/100
      if P>=5001 and P<=10000:
         PRO=0 * 5000/100+(P-5000)*5/100
      if P>10000:
         PRO=0 * 5000/100+5000*5/100 + (P-10000)*10/100
      SX=SX+PRO
   if TY=="Y" :
      if P>=1 and P<=5000:
         PRO=P*2/100
      if P>=5001 and P<=10000:
         PRO=2 * 5000/100+(P-5000)*6/100
      if P>10000:
         PRO=2 * 5000/100+5000*6/100 + (P-10000)*7/100
      SY=SY+PRO
   if TY=="Z" :
      if P>=1 and P<=5000:
         PRO=P*4/100
      if P>=5001 and P<=10000:
         PRO=4 * 5000/100+(P-5000)*6/100
      if P>10000:
         PRO=4 * 5000/100+5000*6/100 + (P-10000)*8/100
      SZ=SZ+PRO

   SP=SP+PRO
   TY=raw_input("Δώσε τον τύπο ενός προϊόντος")

print "προμήθεια X προϊόντος",SX
print "προμήθεια Y προϊόντος",SY
print "προμήθεια Z προϊόντος",SZ

print "συνολική προμήθεια ",SP
```

ΑΣΚΗΣΗ 2

```
HM=[] ;  TA=[]  ; THESI=[]
TAMEIO=[0,0,0,0,0,0,0,0,0,0]
N=input("Dose arithmo asthenon")
while N<0 or N>500:
   N=input("Dose SOSTO arithmo asthenon")

for x in range (N):
   H=input("Dose hmeres nosilias")
   while H<=0:
      H=input("Dose SOSTES hmeres nosilias")
   T=input("Dose to tameio")
   while T<0 or T>10:
      T=input("Dose to SOSTO tameio")
   TH=str(raw_input('Dose thesi nosilias'))
   while TH!="A" and TH!="B" and TH!="C":
      TH=str(raw_input('Dose SOSTH thesi nosilias'))

   HM.append(H)
   TA.append(T)
   THESI.append(TH)
S=0.0
for x in HM:
   S=S+x
MO=S/len(N)
print MO

for x in range(N):
   if THESI[x]=="A":
      Xr=HM[x]*125
   TAMEIO[x]=TAMEIO[x]+Xr
   if THESI[x]=="B":
      Xr=HM[x]*90
   TAMEIO[x]=TAMEIO[x]+Xr
   if THESI[x]=="C":
      Xr=HM[x]*60
   TAMEIO[x]=TAMEIO[x]+Xr
   print Xr

S1=0
for x in range(N):
   print "Tameio ", x+1, "tha plirosei ", TAMEIO[x]
   S1=S1+TAMEIO[x]
print "Ola ta tameia tha plirosoun", S1

for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if TAMEIO[j]>TAMEIO[j-1]:
         TAMEIO[j],TAMEIO[j-1]=TAMEIO[j-1],TAMEIO[j]
         HM[j],HM[j-1]=HM[j-1],HM[j]
         TA[j],TA[j-1]=TA[j-1],TA[j]
         THESI[j],THESI[j-1]=THESI[j-1],THESI[j]
print TA[0], TA[1],TA[2]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2006

1)Σε ένα διαγωνισμό του ΑΣΕΠ εξετάζονται 1500 υποψήφιοι. Ως εξεταστικό κέντρο χρησιμοποιείται ένακτίριο με αίθουσες διαφορετικής χωρητικότητας.

Ο αριθμός των επιτηρητών που απαιτούνται ανάαίθουσα καθορίζεται αποκλειστικά με βάση τηχωρητικότητα της αίθουσας ως εξής:

ΧΩΡΗΤΙΚΟΤΗΤΑ	ΑΡΙΘΜΟΣ ΕΠΙΤΗΡΗΤΩΝ
Μέχρι και 15 θέσεις	1
Από 16 μέχρι και 23 θέσεις	2
Πάνω από 23 θέσεις	3

Να γίνει πρόγραμμα σε γλώσσα προγραμματισμού**python** το οποίο:

α. για κάθε αίθουσα θα διαβάζει τη χωρητικότητά της, θα υπολογίζει και θα εμφανίζει τον αριθμό τωνεπιτηρητών που χρειάζονται. Ο υπολογισμός τουαριθμού των επιτηρητών να γίνεται από συνάρτησηπου θα κατασκευάσετε για το σκοπό αυτό.

β. θα σταματάει όταν εξασφαλισθεί ο απαιτούμενοςσυνολικός αριθμός θέσεων.

Σημείωση: Να θεωρήσετε ότι η συνολική χωρητικότητατων αιθουσών του κτιρίου επαρκεί για τοναριθμό των υποψηφίων.

2) Για την παρακολούθηση των θερμοκρασιών της επικράτειας κατά το μήνα Μάιο καταγράφεται κάθε μέρα η θερμοκρασία στις 12:00 το μεσημέρι για 20 πόλεις. Να σχεδιάσετε αλγόριθμο σε Python που:

α. θα διαβάζει τα ονόματα των 20 πόλεων και θα τα καταχωρεί στην λίστα ΟΝ.

β.θα διαβάζει τις αντίστοιχες θερμοκρασίες για κάθε μία από τις ημέρες του μήνα και θα καταχωρεί τα στοιχεία στη λίστα Τ με κατάλληλο τρόπο (λίστα μέσα σε λίστα).

γ. θα διαβάζει το όνομα μίας πόλης και θα εμφανίζει τη μέγιστη θερμοκρασία της στη διάρκεια του μήνα. Αν δεν υπάρχει η πόλη στην λίστα, θα εμφανίζει κατάλληλα διαμορφωμένο μήνυμα.

δ. θα εμφανίζει το πλήθος των πόλεων που η μέση θερμοκρασία τους ξεπέρασε τους 20^{o} C, αλλά όχι τους 30^{o} C.

ΑΣΚΗΣΗ 1

```
def epitirites(A):
   if A>=1 and A<=15:
      EP=1
   if A>=16 and A<=23:
      EP=2
   if A>23:
      EP=3
   return EP

Y=1500
A=input("Δώσε χωρητικότητα")
whileY>=0:
Y=Y-A
print "Θα χρειαστούν ", epitirites(A), " επιτηρητές"
A=input("Δώσε χωρητικότητα")
```

ΑΣΚΗΣΗ 2

```
ON=[]  ;  T=[]

for x in range(20):
   on=str(raw_input("Dose onoma"))
   ON.append(on)
   B=[]
   for y in range(31):
      t=input("Dose thermokrasia")
      B.append(t)
   T.append(B)

onoma=raw_inpt("Δωσε όνομα")

if onoma in ON:
   for x in range(len(ON)):
      if ON[x]==onoma:
         P=x
   max1=0
   for x in range(31):
      if T [P] [x] > max1:
         max1 = T [P] [x]
   print "μέγιστη θερμοκρασία ", max1
else:
print "Δεν υπάρχει η πόλη"

c=0
for x in range(20):
   S=0.0
   for y in range(31):
      S=S + T [x] [y]
   if S/31 > 20 and S/31 <=30:
      c=c+1

print "Αριθμός πόλεων", c
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2006

1)Σε ένα πάρκινγκ η χρέωση γίνεται κλιμακωτά, όπως φαίνεται στον παρακάτω πίνακα:

ΔΙΑΡΚΕΙΑ ΣΤΑΘΜΕΥΣΗΣ	**ΚΟΣΤΟΣ ΑΝΑ ΩΡΑ**
Μέχρι και 3 ώρες	2 €
Πάνω από 3 έως και 5 ώρες	1,5 €
Πάνω από 5 ώρες	1,3 €

Α. Να κατασκευάσετε πρόγραμμα σε Pythonto οποίο:

α) για κάθε αυτοκίνητο που στάθμευσε στο πάρκινγκ:

I. διαβάζει τον αριθμό κυκλοφορίας μέχρι να δοθείτο 0. Να θεωρήσετε ότι ο αριθμός κυκλοφορίας μπορεί να περιέχει τόσο γράμματα όσο και αριθμούς.
II. διαβάζει τη διάρκεια στάθμευσης σε ώρες και τη δέχεται μόνο εφ' όσον είναι μεγαλύτερη από το 0.
III. καλεί υποπρόγραμμα για τον υπολογισμό του ποσού που πρέπει να πληρώσει ο κάτοχός του.
IV. εμφανίζει τον αριθμό κυκλοφορίας και το ποσό που αναλογεί.

β) εμφανίζει το πλήθος των αυτοκινήτων που έμειναν στο πάρκινγκ μέχρι και δύο ώρες.

Β. Να κατασκευάσετε το υποπρόγραμμα που καλείται στο ερώτημα α) iii.

2) Στους προκριματικούς αγώνες ιππικού τριάθλου συμμετέχουν 16 αθλητές. Τα αγωνίσματα είναι: «ιππική δεξιοτεχνία», «υπερπήδηση εμποδίων» και «ελεύθερη ιππασία». Ο κάθε αθλητής βαθμολογείται ξεχωριστά σε κάθε ένα από τα τρία αγωνίσματα. Να γραφεί πρόγραμμα Python το οποίο:

Α. Να τοποθετεί στη λίστα events τις ονομασίες των τριών αγωνισμάτων, όπως αυτές δόθηκαν παραπάνω.

Β. Να διαβάζει και τοποθετεί στις λίστες name, h_name το όνομα αθλητή και το όνομα του αλόγου κάθε αθλητή.

Γ. Να διαβάζει και τοποθετεί στις λίστες v1, v2 και v3 αντίστοιχα τη βαθμολογία κάθε αθλητή σε κάθε αγώνισμα.

Δ. Να διαβάζει το όνομα ενός αθλητή και να εμφανίζει το όνομα του αλόγου με το οποίο αγωνίστηκε και τη συνολική του βαθμολογία στα τρία αγωνίσματα. Αν δεν υπάρχει ο αθλητής, να εμφανίζει κατάλληλα διαμορφωμένο μήνυμα.

Ε. Να εμφανίζει το όνομα (ή τα ονόματα) του αγωνίσματος (ή των αγωνισμάτων) με το μεγαλύτερο «άνοιγμα βαθμολογίας». Ως «άνοιγμα βαθμολογίας» να θεωρήσετε τη διαφορά ανάμεσα στην καλύτερη και στη χειρότερη βαθμολογία του αγωνίσματος.

ΑΣΚΗΣΗ 1

```
def parking(A):
   if A>=1 and A<=3:
      xr= A*2
   if A>=4 and A<=5:
      xr= 3*2 + (A-3)*1.5
   if A>5:
      xr= 3*2 + 2*1.5 + (A-5)*1.3
   return xr

c=0
AK=raw_input("Δωσε αριθμό κυκλοφορίας")

whileAK!="0":
DS=input("Δωσε διάρκεια στάθμευσης ")
whileDS<=0:
DS=input("Δωσε ΣΩΣΤΗ διάρκεια στάθμευσης ")

print AK, parking(DS)

   if DS <=2:
      c=c+1

AK=raw_input("Δωσε αριθμό κυκλοφορίας")

print "πλήθος αυτοκινήτων ", c
```

ΑΣΚΗΣΗ 2

```
def  MAX_MIN(A):
   max1=A[0]
   min1=A[0]
   for  x  in range(1, len(A)):
      if A[x]> max1:
         max1=A[x]
      if A[x]< min1:
         min1=A[x]
   return max1 - min1

events=[] ; name=[] ; h_name=[] ; V1=[] ; V2=[] ; V3=[] ; D=[]

for  x  in range (3):
   Ag=str(raw_input('Dose onoma agonismatos'))
   events.append(Ag)

for  x  in range(16):
   on=str(raw_input('Dose onoma athliti'))
   on_h==str(raw_input('Dose onoma alogou'))
   name.append(on)
   h_name.append(on_h)
   v1=input('Dose 1h vathmologia')
   v2=input('Dose 1h vathmologia')
   v3=input('Dose 1h vathmologia')
   V1.append(v1)
   V2.append(v2)
   V3.append(v3)

athlitia=raw_input('Dose onoma athliti')
F=False
for  x  in range(16):
   if name[x]== athlitis:
      print name[x], V1[x]+V2[x]+V3[x]
      F=True
if F==False:
   print "Den yparxei o athlitis"

D.append(MAX_MIN(V1))
D.append(MAX_MIN(V2))
D.append(MAX_MIN(V3))

max1=D[0]
if D[1]>max1:
   max1=D[1]
if D[2]>max1:
   max1=D[2]

for x  in range(3):
   if D[x]==max1:
      print events[x], D[x]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2006

1)Οι εκατό(100) υπάλληλοι μιας εταιρείας εργάζονται40 ώρεςτην εβδομάδα. Κάθε ώρα υπερωρίας αμείβεται με5 € (ευρώ).

Να γράψετε αλγόριθμο σε Python ο οποίος:

Α. Για καθένα από τους υπαλλήλους της εταιρείας

α. διαβάζει το όνομά του και για κάθε μέρα από τις πέντε(5) εργάσιμες της εβδομάδας διαβάζει τις ώρεςεργασίας του.

β. υπολογίζει τις εβδομαδιαίες ώρες εργασίας του.

γ. εάν έχει εργαστεί περισσότερο από40 ώρες τηνεβδομάδα, εμφανίζει το όνομά του και υπολογίζει καιεμφανίζει την αμοιβή του για τις υπερωρίες του.

Β. Υπολογίζει και εμφανίζει, στο τέλος, το πλήθος τωνυπαλλήλων που έχουν εργαστεί λιγότερο από40 ώρεςτην εβδομάδα.

2)Για τη διεκδίκηση μιας θέσης υποτροφίας, εξετάστηκαν καιβαθμολογήθηκαν πενήντα(50) υποψήφιοι σε τρία μαθήματα. Ο υπολογισμός του τελικού βαθμού κάθε υποψηφίου γίνεταιως εξής:

Αν ο βαθμός του σε κάποιο από τα τρία μαθήματα είναι μικρότερος του 6, τότε ο τελικός βαθμός του είναι μηδέν (0). Διαφορετικά ο βαθμός του $1^{ου}$μαθήματος συμμετέχει στον υπολογισμό του τελικού βαθμού με συντελεστή 20%, οβαθμός του $2^{ου}$μαθήματος με συντελεστή 35% και ο βαθμός του $3^{ου}$ μαθήματος με συντελεστή 45%.

Να αναπτύξετε αλγόριθμο σε Python ο οποίος:

α. Διαβάζει τα ονόματα των 50 υποψηφίων και τακαταχωρίζει σε λίστα.

β. Διαβάζει για κάθε υποψήφιο τους βαθμούς του σεκαθένα από τα τρία μαθήματα και τους καταχωρίζει σε λίστες, ελέγχοντας ότι ο βαθμός κάθεμαθήματος είναι από 0 έως και 10.

γ. Υπολογίζει τον τελικό βαθμό κάθε υποψηφίου και τονκαταχωρίζει σε λίστα.

δ. Ταξινομεί τα ονόματα και τους τελικούς βαθμούς των υποψηφίων σε φθίνουσα σειρά ως προς τον τελικόβαθμό.

ε. Εμφανίζει για όσους υποψηφίους έχουν τελικό βαθμόμεγαλύτερο του μηδενός (0) το όνομα και τον τελικό βαθμό τους.

στ. Εμφανίζει το ποσοστό των υποψηφίων που έχουν τελικό βαθμό μηδέν (0).

ΑΣΚΗΣΗ 1

```
c=0
for x in range(100):
   on=str(raw_input("Δωσε ονομα"))
   S=0
   for y in range(5):
      ores=input("Δωσε ώρες εργασίας")
S=S+ores

   if S >40:
      print "Ο υπάλληλος με όνομά :",on, "έχει αμοιβή ", S*5

   ifS<40:
c=c+1

print "το  πλήθος που  έχουν εργαστεί λιγότερο από 40 ώρες ", c
```

ΑΣΚΗΣΗ 2

```
ON=[]
VT=[]
for x in range(50):
   on=str(raw_input("Δωσε ονομα"))

   v1=input("Δωσε βαθμό 1")
   while v1<0 or v1>10:
      v1=input("Δωσε ΣΩΣΤΟ βαθμό 1")

v2=input("Δωσε βαθμό 2")
   while v2<0 or v2>10:
      v2=input("Δωσε ΣΩΣΤΟ βαθμό 2")

v3=input("Δωσε βαθμό 3")
   while v3<0 or v3>10:
      v3=input("Δωσε ΣΩΣΤΟ βαθμό 3")

if v1<6 or v2<6 or v3<6:
      Vt=0
   else:
      Vt=v1*20/100 + v2*35/100 + v3*45/100

   ON.append(on)

   VT.append(Vt)

N=len(VT)
for i in range(1,N,1):
   for j in range (N-1,i-1,-1):
      if VT[j]>VT[j-1]:
         VT[j], VT[j-1]=VT[j-1], VT[j]
         ON[j], ON[j-1]=ON[j-1], ON[j]
```

```
c=0
for x in range (len(VT)):
   if VT[x]!=0:
      print ON[x], VT[x]
   else:
c=c+1

print "ποσοστό υποψηφίων με τελικό βαθμό 0 ",
100*c/50
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2006

1) Κάθε ένας αγρότης παράγει ένα μόνο προϊόν από τα δυο που επιδοτούνται. Να γραφεί αλγόριθμος σε Pythono οποίος:

Α) Διαβάζει το ονοματεπώνυμο του αγρότη, το είδος του προϊόντος που παράγει και την ποσότητα του προϊόντος σε κιλά, ελέγχοντας την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:

- Το είδος του προϊόντος είναι Α ή Β.
- Η ποσότητα του προϊόντος είναι θετικός αριθμός.

Β) Υπολογίζει την επιδότηση που δικαιούται ο αγρότης για το είδος του προϊόντος που παράγει.

Η επιδότηση υπολογίζεται κλιμακωτά ανάλογα με την ποσότητα και το είδος του προϊόντος σύμφωνα με τον παρακάτω πίνακα:

Ποσότητα προϊόντος σε κιλά	**Επιδότηση ανά κιλό προϊόντος σε ευρώ**	
	Προϊόν Α	**Προϊόν Β**
έως και 1000	0,8	0,7
από 1001 έως και 2500	0,7	0,6
από 2501 και άνω	0,6	0,5

Γ) Εμφανίζει το ονοματεπώνυμο του αγρότη, το είδος του προϊόντος που παράγει και το ποσό της επιδότησης που δικαιούται.

2) Σε ένα Εσπερινό Γυμνάσιο φοιτούν 80 μαθητές. Να γραφεί αλγόριθμος σε Pythono οποίος:

Α) Διαβάζει για κάθε μαθητή το ονοματεπώνυμό του, την τάξη του και τον τελικό βαθμό του και τα καταχωρεί σε λίστες, ελέγχοντας την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:

- Οι τάξεις είναι Α ή Β ή Γ.
- Ο τελικός βαθμός είναι από 1 μέχρι και 20.

Β) Εμφανίζει τα ονόματα των μαθητών της Β τάξης που έχουν τελικό βαθμό μεγαλύτερο ή ίσο του 18,5.

Γ) Υπολογίζει και εμφανίζει το πλήθος των μαθητών κάθε τάξης.

Δ) Υπολογίζει και εμφανίζει το μέσο όρο των τελικών βαθμών των μαθητών της Γ τάξης.

Ε) Εμφανίζει ταξινομημένα κατά αλφαβητική σειρά τα ονοματεπώνυμα και τους αντίστοιχους τελικούς βαθμούς των μαθητών της Α τάξης.

ΑΣΚΗΣΗ 1

```
on=raw_input("Δώσε όνομα")

E=raw_input("Δώσε είδος προϊόντος")
while E!="A" and E!="B":
   E=raw_input("Δώσε ΣΩΣΤΟ είδος προϊόντος")

K=input("Δώσε ποσότητα προϊόντος ")
whileK<0:
K=input("Δώσε ΣΩΣΤΗ ποσότητα προϊόντος")

if E=="A":
   if K>=1 and K<=1000:
      xr=K*0.8
   if K>=1001 and K<=2500:
      xr=1000*0.8 + (K-1000)*0.7
   if K>2500:
      xr=1000*0.8 + 1500*0.7 + (K-2500)*0.6

if E=="B":
   if K>=1 and K<=1000:
      xr=K*0.7
   if K>=1001 and K<=2500:
      xr=1000*0.7 + (K-1000)*0.6
   ifK>2500:
xr=1000*0.7 + 1500*0.6 + (K-2500)*0.5

print "όνομα: ",on, "είδος προϊόντος: ", E, "επιδότηση :", xr
```

ΑΣΚΗΣΗ 2

```
ON=[]
T=[]
B=[]

for x in range (80):
   on=str(raw_input("Δωσε ονομα"))
   t=str(raw_input("Δωσε την τάξη"))
while t not in ["A","B","Γ"]:
      t=str(raw_input("Δωσε ΣΩΣΤΗ την τάξη"))
v=input("Δωσε τον βαθμό")
while v <0 or v>20:
      v=input("Δωσε ΣΩΣΤΟ βαθμό")
   ON.append(on)
   T.append(t)
   B.append(v)

for x in range(len(B)):
   if B[x]>=18.5 and T[x]=="B":
      print ON[x]

ca=0
cb=0
cc=0
for x in T:
   if x=="A":
      ca=ca+1
   if x=="B":
      cb=cb+1
   if x=="Γ":
      cc=cc+1

print "μαθητές Α Τάξης",ca
print "μαθητές Β Τάξης",cb
print "μαθητές Γ Τάξης",cc

S=0.0
c=0
for x in range(len(B)):
   if T[x]=="Γ":
      S=S+B[x]
      c=c+1

print "μέσος όρος μαθητών Γ τάξης", S/c
```

```
A=[]
AT=[]
for x in range(80):
   if T[x]=="A":
      A.append(ON[x])
      AT.append(B[x])

N=len(A)
for i in range(1,N,1):
   for j in rnage(N-1,i-1,-1):
      if A[j]<A[j-1]:
         A[j], A[j-1]= A[j-1], A[j]
         AT[j], AT[j-1]= AT[j-1], AT[j]

for x  in range(len(A)):
   print A[x], AT[x]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2007

1) Ένας συλλέκτης γραμματοσήμων επισκέπτεται στοδιαδίκτυο το αγαπημένο του ηλεκτρονικό κατάστημαφιλοτελισμού προκειμένου να αγοράσει γραμματόσημα. Προτίθεται να ξοδέψει μέχρι 1500 ευρώ.

Να αναπτύξετε αλγόριθμο σε Pythono οποίος:

α. Για κάθε γραμματόσημο, να διαβάζει την τιμή και τηνπροέλευσή του(ελληνικό/ξένο) και να επιτρέπει την αγορά του, εφόσον η τιμή του δεν υπερβαίνει τοδιαθέσιμο υπόλοιπο χρημάτων. Διαφορετικά νατερματίζει τυπώνοντας το μήνυμα«ΤΕΛΟΣ ΑΓΟΡΩΝ».

ΣΗΜΕΙΩΣΗ: Δεν απαιτείται έλεγχος εγκυρότηταγια τα δεδομένα εισόδου.

β. Να τυπώνει:

1. Το συνολικό ποσό που ξόδεψε ο συλλέκτης.

2. Το πλήθος των ελληνικών και το πλήθος τωνξένων γραμματοσήμων που αγόρασε.

3. Το ποσό που περίσσεψε, εφόσον υπάρχει, διαφορετικά το μήνυμα«ΕΞΑΝΤΛΗΘΗΚΕ ΟΛΟΤΟ ΠΟΣΟ».

2) Μια δισκογραφική εταιρεία καταγράφει στοιχεία γιαένα έτος για κάθε ένα από τα 20 CDs που κυκλοφόρησε. Τα στοιχεία αυτά είναι ο τίτλος του CD, ο τύπος της μουσικής που περιέχει και οι εξαμηνιαίες του πωλήσεις(ποσά σε ευρώ) στη διάρκεια του έτους. Οι τύποι μουσικής είναι δύο: «ορχηστρική» και«φωνητική».

Να αναπτυχθεί αλγόριθμος σε Pythono οποίος:

α. Για κάθε ένα από τα20 CDs, να διαβάζει τον τίτλο, τον τύπο της μουσικής και τις πωλήσεις του για κάθε εξάμηνο, ελέγχοντας την έγκυρη καταχώριση του τύπουτης μουσικής.

β. Να εμφανίζει τον τίτλο ή τους τίτλους των CDs με τις περισσότερες πωλήσεις τον $1^{ου}$ εξάμηνου του έτους.

γ. Να εμφανίζει τους τίτλους των ορχηστρικών CDs μεετήσιο σύνολο πωλήσεων τουλάχιστον 5000 ευρώ.

δ. Να εμφανίζει πόσα από τα CDs είχαν σύνολοπωλήσεων στο δεύτερο εξάμηνο μεγαλύτερο απ' ό,τιστο πρώτο.

ΑΣΚΗΣΗ 1

```
P=1500
c1=0
c2=0

T=input("Δωσε τιμή")

whileT<=P:
Pr=raw_input("Δωσε χώρα προέλευσής")
if Pr=="Ελλαδα":
      c1=c1+1
   if Pr=="Ξένο":
      c2=c2+1

   T=input("Δωσε τιμή")

print "ΤΕΛΟΣ ΑΓΟΡΩΝ"
print "συνολικό ποσό που ξόδεψε", 1500-P
print "πλήθος  ελληνικών  ", c1
print "πλήθος  ξένων", c2

ifP!=0:
print "ποσό  που  περίσσεψε", P
else:
print "ΕΞΑΝΤΛΗΘΗΚΕ  ΟΛΟ ΤΟ ΠΟΣΟ"
```

ΑΣΚΗΣΗ 2

```
T=[]
TY=[]
EX1=[]
EX2=[]

for x in range(20):
   t=str(raw_input("Δώσε τίτλο"))
   ty=str(raw_input("Δώσε τύπο"))
   while ty!="ορχηστρική" and ty!="φωνητική":
      ty=str(raw_input("Δώσε τύπο"))

   ex1=input("Δώσε πωλήσεις 1 εξάμηνου ")
ex2=input("Δώσε πωλήσεις 2 εξάμηνου ")

T.append(t)
   TY.append(ty)
   EX1.append(ex1)
   EX2.append(ex2)

max1=0
for x in EX1:
   if x> max1:
      max1=x

for x in range(len(EX1)):
   if EX1[x]==max1:
      print T[x]

for x in range(len(EX1)):
   if EX1[x]+EX2[x]>=5000:
      print T[x]

c=0
for x in range(len(EX1)):
   if EX2[x]>EX1[x]:
      c=c+1

print "πωλήσεις δεύτερου εξάμηνου μεγαλύτερες από πρώτου", c
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2007

1)Το κλασικό παιχνίδι«Πέτρα – Ψαλίδι-Χαρτί» παίζεται με δύοπαίκτες. Σε κάθε γύρο του παιχνιδιού, ο κάθε παίκτηςεπιλέγει ένα από τα ΠΕΤΡΑ, ΨΑΛΙΔΙ, ΧΑΡΤΙ, καιπαρουσιάζει την επιλογή του ταυτόχρονα με τον αντίπαλότου. Η ΠΕΤΡΑ κερδίζει το ΨΑΛΙΔΙ, το ΨΑΛΙΔΙ το ΧΑΡΤΙκαι το ΧΑΡΤΙ την ΠΕΤΡΑ. Σε περίπτωση που οι δύο παίκτεςέχουν την ίδια επιλογή, ο γύρος λήγει ισόπαλος. Το παιχνίδιπροχωράει με συνεχόμενους γύρους μέχρι ένας τουλάχιστοναπό τους παίκτες να αποχωρήσει. Νικητής αναδεικνύεται οπαίκτης με τις περισσότερες νίκες. Αν οι δύο παίκτες έχουντον ίδιο αριθμό νικών, το παιχνίδι λήγει ισόπαλο. Να αναπτύξετε αλγόριθμο σε pythono οποίος διαβάζει τα ονόματα τωνδύο παικτών και υλοποιεί το παραπάνω παιχνίδι ως εξής:

Α. Για κάθε γύρο του παιχνιδιού:

1. διαβάζει την επιλογή κάθε παίκτη, η οποία μπορεί ναείναι μία από τις εξής: ΠΕΤΡΑ, ΨΑΛΙΔΙ, ΧΑΡΤΙ, ΤΕΛΟΣ. (Δεν απαιτείται έλεγχος εγκυρότητας τιμών.)

2. συγκρίνει τις επιλογές των παικτών και διαπιστώνειτο νικητή του γύρου ή την ισοπαλία.

Β. Τερματίζει το παιχνίδι όταν ένας τουλάχιστον από τους δύο παίκτες επιλέξει ΤΕΛΟΣ.

Γ. Εμφανίζει το όνομα του νικητή ή, αν δεν υπάρχεινικητής, το μήνυμα«ΤΟ ΠΑΙΧΝΙΔΙ ΕΛΗΞΕ ΙΣΟΠΑΛΟ».

2) Μια σύγχρονη πτηνοτροφική μονάδα παρακολουθεί τηνημερήσια παραγωγή αυγών και καταγράφει τα στοιχεία σε ηλεκτρονικό αρχείο. Να αναπτύξετε αλγόριθμο ο οποίος θα διαχειρίζεται τα στοιχεία της μονάδας στη διάρκεια ενός έτους. Για το σκοπό αυτό:

Α. Να κατασκευάσετε κύριο πρόγραμμα το οποίο:

1. να δημιουργεί λίστα **ON** με τα ονόματα των μηνών ενός έτους

2. να ζητάει το έτος παρακολούθησης, ελέγχοντας ότι πρόκειται για έτος του21ου αιώνα (από 2000 μέχρι και 2099). Ο αλγόριθμος να δημιουργεί λίστα **ΜΗ** με τον αριθμό των ημερών για καθέναν από τους δώδεκαμήνες του έτους που δόθηκε. Ο αριθμός των ημερώντου μήνα θα υπολογίζεται από υποπρόγραμμα τοοποίο θα κατασκευάσετε για το σκοπό αυτό. Ηλειτουργία του υποπρογράμματος περιγράφεται στο ερώτημα Β.

3. Για κάθε μήνα να ζητάει την ημερήσια παραγωγή (αριθμό αυγών) και να υπολογίζει και να καταχωρίζει το σύνολο των αυγών του μήνα στην λίστα **SUM**.

4. να εμφανίζει το όνομα του μήνα με μηνιαία παραγωγή αυγώνπλησιέστερα στον ετήσιο μέσο όρο παραγωγής.

Β. Να κατασκευάσετε υποπρόγραμμα το οποίο να δέχεταιως παραμέτρους κάποιο έτος και τον αριθμό κάποιουμήνα(1 έως12), και να επιστρέφει τον αριθμό τωνημερών του συγκεκριμένου μήνα. Όταν το έτος είναιδίσεκτο, ο Φεβρουάριος έχει29 ημέρες, διαφορετικά έχει28. Δίσεκτα είναι τα έτη που διαιρούνται με το4 αλλάόχι με το100, καθώς και εκείνα που διαιρούνται με το400. Για τους υπόλοιπους μήνες, πλην του Φεβρουαρίου,ισχύει το εξής: μέχρι και τον Ιούλιο($7^{ος}$μήνας) οι μονοίμήνες έχουν31 ημέρες και οι ζυγοί30. Για τους μήνεςμετά τον Ιούλιο, ισχύει το αντίστροφο.

ΑΣΚΗΣΗ 1

```
ON1=raw_input("Δωσε ονομα παικτη Α")
ON2=raw_input("Δωσε ονομα παικτη Β")

E1=raw_input("Δώσε επιλογή παικτη Α")
E2=raw_input("Δώσε επιλογή παικτη Β")

c1=0
c2=0
while E1!="ΤΕΛΟΣ" and E2!="ΤΕΛΟΣ":

   if E1==E2:
      print "ισοπαλία"
   elif E1=="ΠΕΤΡΑ" :
      if E2=="ΨΑΛΙΔΙ" :
         print "κερδίζει  ο Α"
         c1+=1
      if E2=="ΧΑΡΤΙ" :
         print "κερδίζει  ο Β"
         c2+=1
   elif E1=="ΨΑΛΙΔΙ" :
      if E2=="ΠΕΤΡΑ" :
         print "κερδίζει  ο Β"
         c2+=1
      if E2=="ΧΑΡΤΙ" :
         print "κερδίζει  ο Α"
         c1+=1
   elif E1=="ΧΑΡΤΙ" :
      if E2=="ΠΕΤΡΑ" :
         print "κερδίζει  ο Α"
         c1+=1
      if E2=="ΨΑΛΙΔΙ" :
         print "κερδίζει  ο Β"
         c2+=1

E1=raw_input("Δώσε επιλογή παικτη Α")
E2=raw_input("Δώσε επιλογή παικτη Β")

if c1>c2:
   print "νικητή  ο ", ON1
if c2>c1:
   print "νικητή  ο ", ON2
ifc1==c2:
print " ΤΟ ΠΑΙΧΝΙΔΙ ΕΛΗΞΕ ΙΣΟΠΑΛΟ "
```

ΑΣΚΗΣΗ 2

```
def  MHNAS(E,M):
   if (E%4==0 and E%100!=0) or E%400==0:
      if M==2:
         return 29
   else:
      if M==2:
         return 28
   if M<=7:
      if M%2==1:
         return 31
      if M%2==0:
         return 30
   elif M>7:
      if M%2==1:
         return 30
      if M%2==0:
return 31

ON=["Ιανουάριος", "Φεβρουάριος","Μάρτιος","Απρίλιος","Μάιος", "Ιούνιος", "Ιούλιος", "Αυγουστος",
"Σεπτέμβριος", "Οκτώμβριος","Νοέμβριος", "Δεκέμβριος"]
MH=[]
SUM=[]

E=input("Δώσε έτος")
for x in range(12):
   MH.append(MHNAS(E,x+1))

for x in range(12):
   S=0
   for y in range(MH[x]):
P=input("Δώσε παραγωγή αυγών")
S=S+P
   SUM.append(S)

S=0.0
for x in SUM:
   S=S+x
MO=S/12

min1=SUM[0]
for x in range(len(SUM)):
   if MO>SUM[x]:
      if MO-SUM[x] < min1:
         min1 = MO-SUM[x]
         Y=x
   else:
      if ads(SUM[x]-MO) < min1:
         min1 = abs(SUM[x]-MO)
Y=x
print ON[Y]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2007

1) Μία εταιρεία ασφάλισης οχημάτων καθορίζει το ετήσιο κόστος ασφάλισης ανά τύπο οχήματος (δίκυκλο ή αυτοκίνητο) και κυβισμό, σύμφωνα με τους παρακάτω πίνακες:

ΔΙΚΥΚΛΟ	
Κυβισμός (σε κυβικά εκατοστά)	**Κόστος Ασφάλισης (σε ευρώ)**
έως και 125	100
πάνω από 125	140

ΑΥΤΟΚΙΝΗΤΟ	
Κυβισμός (σε κυβικά εκατοστά)	**Κόστος Ασφάλισης (σε ευρώ)**
έως και 1400	400
από 1401 έως και 1800	500
πάνω από 1800	700

Αν η ηλικία του οδηγού είναι από 18 έως και 24 ετών τότε το κόστος της ασφάλισης του οχήματος προσαυξάνεται κατά 10%.

Να αναπτύξετε αλγόριθμο σε Python, ο οποίος:

α. Να διαβάζει την ηλικία ενός οδηγού, τον τύπο του οχήματος και τον κυβισμό του, ελέγχοντας ώστε ο τύπος του οχήματος να είναι «ΔΙΚΥΚΛΟ» ή «ΑΥΤΟΚΙΝΗΤΟ».

β. Να υπολογίζει και να εμφανίζει το ετήσιο κόστος ασφάλισης του οχήματος.

Σημείωση: Να θεωρήσετε ότι η ηλικία του οδηγού είναι τουλάχιστον 18 ετών.

2) Σε ένα πανεπιστημιακό τμήμα εισήχθησαν κατόπιν γενικών εξετάσεων 235 φοιτητές προερχόμενοι από την ΤΕΧΝΟΛΟΓΙΚΗ ή τη ΘΕΤΙΚΗ κατεύθυνση.

Να αναπτύξετε πρόγραμμα σε Pythonto οποίο:

α. Για καθένα από τους 235 φοιτητές διαβάζει:

- το ονοματεπώνυμό του,
- τα μόρια εισαγωγής του,
- την κατεύθυνσή του, η οποία μπορεί να είναι «ΤΕΧΝΟΛΟΓΙΚΗ» ή «ΘΕΤΙΚΗ», ελέγχοντας την εγκυρότητα εισαγωγής της και καταχωρίζει τα δεδομένα αυτά σε τρεις λίστες.

β. Υπολογίζει και εμφανίζει:

1. το μέσο όρο των μορίων εισαγωγής των φοιτητών που προέρχονται από την ΤΕΧΝΟΛΟΓΙΚΗ κατεύθυνση.

2. το ποσοστό των φοιτητών, που προέρχονται από τηνΤΕΧΝΟΛΟΓΙΚΗ κατεύθυνση.

3. την κατεύθυνση, από την οποία προέρχεται ο φοιτητής με τα περισσότερα μόρια εισαγωγής (να θεωρήσετε ότι δεν υπάρχει περίπτωση ισοβαθμίας).

4. τα ονοματεπώνυμα των φοιτητών που προέρχονται από την ΤΕΧΝΟΛΟΓΙΚΗ κατεύθυνση, για τους οποίους τα μόρια εισαγωγής τους είναι περισσότερα από το μέσο όρο των μορίων εισαγωγής των φοιτητών που προέρχονται από την ΤΕΧΝΟΛΟΓΙΚΗ κατεύθυνση.

ΑΣΚΗΣΗ 1

```
H=input("Δωσε ηλικία")

T=raw_input("Δωσε τύπο")
while K!="ΔΙΚΥΚΛΟ" and K!="ΑΥΤΟΚΙΝΗΤΟ" :
   T=input("Δώσε ΣΩΣΤΟ τύπο")

K=input("Δώσε κυβισμό ")

if T=="ΔΙΚΥΚΛΟ" :
if K>=1 and K<=125:
      Xr=100
   if K>125:
      Xr=140

if T=="ΑΥΤΟΚΙΝΗΤΟ" :
   if K>=1 and K<=1400:
      Xr=400
   if K>=1401 and K<=1800:
      Xr=500
   if K>1800:
      Xr=700

if H>=18 and H<=24:
Xr=Xr+ Xr*10/100

print "ετήσιο κόστος ασφάλισης", Xr
```

ΑΣΚΗΣΗ 2

```
ON=[]
M=[]
K=[]

for x in range(235):
   on=str(raw_input("Δωσε ονομα"))
   m=input("Δωσε μόρια")
   k==str(raw_input("Δωσε κατεύθυνσή "))
whilek!="ΤΕΧΝΟΛΟΓΙΚΗ"andk!="ΘΕΤΙΚΗ" :
k==str(raw_input("Δωσε ΣΩΣΤΗ κατεύθυνσή "))

ON.append(on)
   M.append(m)
   K.append(K)

S=0.0
c=0
for x in range(235):
   if K[x]=="ΤΕΧΝΟΛΟΓΙΚΗ":
      S=S+M[x]
      c=c+1

MOT=S/c
print "μέσο όρο μορίων εισαγωγής ΤΕΧΝΟΛΟΓΙΚΗΣ", MOT

print "ποσοστό των φοιτητών ΤΕΧΝΟΛΟΓΙΚΗΣ", c*100/235

max1=0
for x in range(235):
   if M[x]> max1:
      max1=M[x]
      Y=K[x]

print "κατεύθυνση με περισσότερα μόρια εισαγωγής ", Y

for x in range(235):
   if K[x]=="ΤΕΧΝΟΛΟΓΙΚΗ" and M[x]>MOT:
      print ON[x]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2007

1) Ένας καταναλωτής διαθέτει150€ για αγορά ρυζιού, προκειμένου να το δωρίσει σε ένα φιλανθρωπικό ίδρυμα. Σε ένα πολυκατάστημα διατίθενται πακέτα ρυζιού σε τέσσερις διαφορετικές συσκευασίες από διαφορετικές εταιρείες.

Να γράψετε αλγόριθμο σε Pythono οποίος:

α. Διαβάζει το όνομα της εταιρείας, την αξία και την ποσότητα σε γραμμάρια για κάθε μία από τις τέσσερις συσκευασίες ρυζιού Α, Β, Γ ή Δ.

β. Υπολογίζει και εμφανίζει το όνομα της εταιρείας που προσφέρει το ρύζι στην πλέον συμφέρουσα για τον καταναλωτή συσκευασία (να θεωρήσετε ότι υπάρχει μόνο μία τέτοια εταιρεία).

γ. Υπολογίζει και εμφανίζει τον αριθμό των πακέτων που μπορεί να αγοράσει από την πλέον συμφέρουσα για τον καταναλωτή συσκευασία (σύμφωνα με το ερώτημα β).

2)Σε ένα Μετεωρολογικό Σταθμό καταγράφονται ανά ημέραη θερμοκρασία του περιβάλλοντος για έναν μήνα.

Να γράψετε αλγόριθμο σε Pythonπου:

α. Διαβάζει:

• το όνομακάθεημέραςνακαταχωρείτε σε λίστα ΟΝ. Ο μήνας ξεκινάει με την ημέρα Δευτέρα και έχει 30 ημέρες.

• τη θερμοκρασία για κάθε ημέρα την καταχωρεί σελίστα Τ, ελέγχοντας οι τιμές της θερμοκρασίας να είναι από20 μέχρι και50.

β. Υπολογίζει για κάθε εβδομάδα τη μέση θερμοκρασία και τηνκαταχωρεί σε λίσταΜΤ.

γ. Βρίσκει και εμφανίζει τη μέγιστη θερμοκρασία του μήνα.

δ. Βρίσκει και εμφανίζει την ημέρα κάθε εβδομάδας με τημέγιστη θερμοκρασία(να θεωρήσετε ότι υπάρχειμόνο μία τέτοια ημέρα).

ε. Υπολογίζει και εμφανίζει το πλήθος των ημερών του μήνα που είχαν θερμοκρασία μεγαλύτερη των20^{o}C.

Σημείωση: Κάθε μήνα με 30 ημέρες έχουμε 4 εβδομάδες και μια 5^{η} με 2 μόνο ημέρες

ΑΣΚΗΣΗ 1

```
POSO=150

E=[]
T=[]
G=[]
S=[]

for x in range(4):
   e=str(raw_input("Δωσε ονομα"))
   g=float(input("Δωσε γραμμαρια"))
   t=float(input("Δωσε τιμή"))
   s=str(raw_input("Δωσε συσκευασία "))

   E.append(e)
   T.append(t)
   G.append(g)
   S.append(s)

min1=T[0]/G[0]
Y=x
for x  in range(1,len(T)):
   if T[x]/G[x] < min1:
      min1= T[x]/G[x]
      Y=x

print "συμφέρουσα  συσκευασία ", E[Y]

print "αριθμός  πακέτων ", POSO//T[Y]
```

ΑΣΚΗΣΗ 2

```
ON=[]
T=[]
MT=[]

for x in range(30):
   on=str(raw_input("Δωσε ημέρα"))
   t=input("Δωσε θερμοκρασία")
   while t<20 or t>50:
      t=input("Δωσε ΣΩΣΤΗ θερμοκρασία")
ON.append(on)
   T.append(t)

for x in range(0, len(T),7):
   S=0.0
   for y in range(7):
      S=S+ T[x+y]

   MT.append(S/7)

MT.append((T[29]+T[30])/2)

max1=0
for x in T:
   if x>max1:
      max1=x

print "μέγιστη θερμοκρασία ", max1

c=1
for x in range(0, len(T),7):
   max1=0
   for y in range(7):
      if T[x+y] > max1:
         max1=T[x+y]
         Z=x+y

print "ημέρα", c," εβδομάδας με μέγιστη θερμοκρασία", ON[Z]
c=c+1

ifT[29]>T[30]:
print "ημέρα ",5," εβδομάδας με μέγιστη θερμοκρασία", ON[29]
else:
print "ημέρα ",5," εβδομάδας με μέγιστη θερμοκρασία", ON[30]

c=0
for x  in T:
   if x>20:
      c=c+1

print"πλήθος  ημερών T>20 ", c
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2008

1) Μία εταιρεία ενοικίασης αυτοκινήτων έχει νοικιάσει30 αυτοκίνητα τα οποία κατηγοριοποιούνται σε οικολογικά και συμβατικά. Η πολιτική χρέωσης για την ενοικίαση ανά κατηγορία και ανά ημέρα δίνεται στον παρακάτω πίνακα.

ΗΜΕΡΕΣ	**ΟΙΚΟΛΟΓΙΚΑ**	**ΣΥΜΒΑΤΙΚΑ**
1-7	30€ ανά ημέρα	40€ ανά ημέρα
8-16	20€ ανά ημέρα	30€ ανά ημέρα
από17 και άνω	10€ ανά ημέρα	20€ ανά ημέρα

1. Να αναπτύξετε πρόγραμμα σε Pythonτο οποίο:

α. Για κάθε αυτοκίνητο το οποίο έχει ενοικιαστεί:

i. Διαβάζει την κατηγορία του(«ΟΙΚΟΛΟΓΙΚΑ» ή«ΣΥΜΒΑΤΙΚΑ») και τις ημέρες ενοικίασης.

ii. Καλεί συνάρτηση με είσοδο την κατηγορίατου αυτοκινήτου και τις ημέρες ενοικίασης καιυπολογίζει με βάση τον παραπάνω πίνακα τη χρέωση.

iii. Εμφανίζει το μήνυμα"χρέωση" και τη χρέωση πουυπολογίσατε.

β. Υπολογίζει και εμφανίζει το πλήθος των οικολογικώνκαι των συμβατικών αυτοκινήτων.

2. Να κατασκευάσετε την κατάλληλησυνάρτηση τουερωτήματος 1.α.ii.

ΣΗΜΕΙΩΣΗ:

1)Δεν απαιτείται έλεγχος εγκυρότητας για ταδεδομένα εισόδου και

2) Ο υπολογισμός της χρέωσης δεν πρέπει ναγίνει κλιμακωτά.

2) Στο ευρωπαϊκό πρωτάθλημα ποδοσφαίρου συμμετέχουν16 ομάδες. Κάθε ομάδα συμμετέχει σε30 αγώνες. Να γράψετεαλγόριθμο σε Pythono οποίος:

α. Διαβάζει και καταχωρεί σε λίστα ΟΝ τα ονόματατων ομάδων.

β. Διαβάζει και καταχωρεί σε λίστεςΑ τααποτελέσματα σε κάθε αγώνα ως εξής:

Τον χαρακτήρα«Ν»για ΝΙΚΗ

Τον χαρακτήρα«Ι» για ΙΣΟΠΑΛΙΑ

Τον χαρακτήρα«Η» για ΗΤΤΑ

και κάνει τον απαραίτητο έλεγχο εγκυρότητας τωνδεδομένων.

γ. Για κάθε ομάδα υπολογίζει και καταχωρεί σε 3 λίστες Ν, Ι, Η το πλήθος των νικών, το πλήθος των ισοπαλιών και το πλήθος των ηττών.

δ. Με βάση τα στοιχεία των λιστών Ν, Ι και Η υπολογίζεικαι καταχωρεί σε νέα λίσταΒ τη συνολικήβαθμολογία κάθε ομάδας, δεδομένου ότι για κάθε νίκη ηομάδα παίρνει τρεις βαθμούς, για κάθε ισοπαλία ένανβαθμό και για κάθε ήττα κανέναν βαθμό.

ΑΣΚΗΣΗ 1

```
defΧREOSH(K,H):

ifK=="ΟΙΚΟΛΟΓΙΚΑ":
if H>=1 and H<=7:
          Xr=K*30
       if H>=8 and H<=16:
          Xr=K*20
       if H>=17:
          Xr=K*10
    if K=="ΣΥΜΒΑΤΙΚΑ":
       if H>=1 and H<=7:
          Xr=K*40
       if H>=8 and H<=16:
          Xr=K*30
       if H>=17:
          Xr=K*20
    return  Xr


c1=0
c2=0
for x  in range(30):
    K=raw_input("Δώσε κατηγορία")
H=input("Δώσε ημέρες ενοικίασης")

print "χρέωση", XREOSH(K,H)

    ifK=="ΟΙΚΟΛΟΓΙΚΑ":
c1=c1+1
ifK=="ΣΥΜΒΑΤΙΚΑ":
c2=c2+1

print "πλήθος οικολογικών", c1
print "πλήθος συμβατικών ", c2
```

ΑΣΚΗΣΗ 2

```
ON=[]
A=[]
N=[]
I=[]
H=[]
B=[]

for x in range(16):
   T=[]
   on=str(raw_input("Δώσε ονομα"))
   ON.append(on)
   for y in range(30):
      a=str(raw_input("Δώσε αποτελέσμα"))
while a not in ["N", "I", "H"]:
        a=str(raw_input("Δώσε N ή I ή H"))
T.append(a)
   A.append(T)

c1=0
c2=0
c3=0
for x in range(16):
   for y in range(30):

      if A[x] [y]=="N":
         c1=c1+1
      if A[x] [y]=="I":
         c2=c2+1
      if A[x] [y]=="H":
         c3=c3+1
   N.append(c1)
   I.append(c2)
   H.append(c3)

for x in range(16):
   B.append(N[x]*3 + I[x])
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2008

1)Μία εταιρεία αποφάσισε να δώσει βοηθητικό επίδομα στους υπαλλήλους της για τον μήνα Ιούλιο. Το επίδομα διαφοροποιείται, ανάλογα με το φύλο του/της υπαλλήλου και τον αριθμό των παιδιών του/της, με βάση τους παρακάτω πίνακες:

ΑΝΔΡΕΣ	
ΑΡΙΘΜΟΣΠΑΙΔΙΩΝ	**ΕΠΙΔΟΜΑ ΣΕ €**
1	20
2	50
>=3	120

ΓΥΝΑΙΚΕΣ	
ΑΡΙΘΜΟΣΠΑΙΔΙΩΝ	**ΕΠΙΔΟΜΑ ΣΕ €**
1	30
2	80
>=3	160

Να γράψετε πρόγραμμα σε Pythonτο οποίο

α. διαβάζει το φύλο («Α» ή «Γ») το οποίο ελέγχεται ως προς την ορθότητα της εισαγωγής του. Επίσης διαβάζει τον μισθό και τον αριθμό των παιδιών του υπαλλήλου.

β. υπολογίζει και εμφανίζει το επίδομα και το συνολικό ποσό που θα εισπράξει ο υπάλληλος τον μήνα Ιούλιο.

γ. δέχεται απάντηση «ΝΑΙ» ή «ΟΧΙ» για τη συνέχεια ή τον τερματισμό της επανάληψης μετά την εμφάνιση σχετικού μηνύματος.

δ. υπολογίζει και εμφανίζει το συνολικό ποσό επιδόματος που πρέπει να καταβάλει η Εταιρεία στους υπαλλήλους της.

2)Στο άθλημα των 110 μέτρων μετ’ εμποδίων, στους δύο ημιτελικούς αγώνες συμμετέχουν δέκα έξι (16) αθλητές (8 σε κάθε ημιτελικό). Σύμφωνα με τον κανονισμό στον τελικό προκρίνεται ο πρώτος αθλητής κάθε ημιτελικού. Η οκτάδα του τελικού συμπληρώνεται με τους αθλητές που έχουν τους έξι (6) καλύτερους χρόνους απ’ όλους τους υπόλοιπους συμμετέχοντες. Να θεωρήσετε ότι δεν υπάρχουν αθλητές με ίδιους χρόνους.

Α. Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο

1. καλεί τη συνάρτηση ΕΙΣΟΔΟΣ για κάθε ημιτελικό ξεχωριστά. Η διαδικασία διαβάζει το όνομα του αθλητή και τον χρόνο του (με ακρίβεια δεκάτου του δευτερολέπτου).

2. καλεί τη συνάρτησηΤΑΞΙΝΟΜΗΣΗ για κάθε ημιτελικό ξεχωριστά. Η διαδικασία ταξινομεί τους αθλητές ως προς τον χρόνο τους με αύξουσα σειρά.

3. δημιουργεί την λίστα ΟΝ με τα ονόματα και την λίστα ΧΡ με τους αντίστοιχους χρόνους των αθλητών που προκρίθηκαν στον τελικό.

4. εμφανίζει τα ονόματα και τους χρόνους των αθλητών που θα λάβουν μέρος στον τελικό.

Β. Να γράψετε τη συνάρτηση ΕΙΣΟΔΟΣ και τη συνάρτηση ΤΑΞΙΝΟΜΗΣΗ.

ΑΣΚΗΣΗ 1

```
ep=raw_input("Θέλετε να δώσετε στοιχεια υπαλλήλου;")
while ep!="ΝΑΙ" and ep!="ΟΧΙ":
   ep=raw_input("δώσετε ΝΑΙ ή ΟΧΙ ")

S=0

while ep =="ΝΑΙ":

   F=raw_input("Δώστε φύλο ")
   while F!="Α" and F!="Γ":
      F=raw_input("Δώστε ΣΩΣΤΟ φύλο ")

M=input("Δώστε μισθό ")

P=input("Δώστε αριθμό παιδιών ")

if F=="Α":
      if P==1:
         TM=M+20
         S=S+20
      if P==2:
         TM=M+50
         S=S+50
      if P>=3:
         TM=M+120
         S=S+120
   if F=="Γ":
      if P==1:
         TM=M+30
         S=S+30
      if P==2:
         TM=M+80
         S=S+80
      if P>=3:
         TM=M+160
         S=S+160

print "συνολικό ποσό που θα εισπράξει ", TM


ep=raw_input("Θέλετε να δώσετε στοιχεια υπαλλήλου;")
while ep!="ΝΑΙ" and ep!="ΟΧΙ":
      ep=raw_input("δώσετε ΝΑΙ ή ΟΧΙ ")

print "συνολικό ποσό επιδόματος ", S
```

ΑΣΚΗΣΗ 2

```
ON1=[]
XR1=[]
ON2=[]
XR2=[]

def BBS(ONt, XRt):
   N=8
   for i in range(1,N,1):
      for j in range(N-1,i-1,-1):
         if XRt[j]<XRt[j-1]:
            XRt[j], XRt[j-1]=XRt[j-1],XRt[j]
            ONt[j], ONt[j-1]=ONt[j-1],ONt[j]

def Eisodos(ONt,XRt):
   on=str(raw_input("Δωσε ονομα"))
   xr=float(input("Δωσε χρόνο"))
   ONt.append(on)
   XRt.append(xr)

ON=[]
XR=[]

for x in range (8):
   print "1ος ημιτελικός "
   Eisodos(ON1,XR1)

BBS(ON1, XR1)

for x in range (8):
   print "2ος ημιτελικός "
   Eisodos(ON2,XR2)

BBS(ON2, XR2)

if XR1[0]<XR2[0] :
   XR.append(XR1[0])
   ON.append(ON1[0])
   XR.append(XR2[0])
   ON.append(ON2[0])
   XR1.pop(0)
   ON1.pop(0)
   XR2.pop(0)
   ON2.pop(0)
else:
   XR.append(XR2[0])
   ON.append(ON2[0])
   XR.append(XR1[0])
   ON.append(ON1[0])
   XR2.pop(0)
   ON2.pop(0)
   XR1.pop(0)
   ON1.pop(0)
```

```
c=1

while c<=6:

   if XR1[0]<XR2[0]:
      XR.append(XR1[0])
      ON.append(ON1[0])
      XR1.pop(0)
      ON1.pop(0)
   else:
      XR.append(XR2[0])
      ON.append(ON2[0])
      XR2.pop(0)
      ON2.pop(0)
   c=c+1

for x  in range(8):
   print ON[x], XR[x]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2008

1) Για την ανάδειξη του επταμελούς(7) ΔιοικητικούΣυμβουλίου ενός Πολιτιστικού Συλλόγου υπάρχουν20 υποψήφιοι.

Να γράψετε αλγόριθμο σε Python ο οποίος

α. διαβάζει τα ονόματα των υποψηφίων και τα αποθηκεύεισε λίστα.

β. διαβάζει για κάθε υποψήφιο τον αριθμό των ψήφων πουέλαβε και τον αποθηκεύει σε λίστα.

γ. εμφανίζει τα ονόματα των εκλεγέντων μελών του Διοικητικού Συμβουλίου κατά φθίνουσα σειρά ψήφων(να θεωρηθεί ότι δεν υπάρχουν περιπτώσεις ισοψηφίας).

δ. διαβάζει το όνομα ενός υποψηφίου και ελέγχει αν οσυγκεκριμένος εκλέγεται ή όχι, εμφανίζοντας κατάλληλο μήνυμα.

2) Ένας επενδυτής διέθεσε10.000 € για την αγορά ορισμένωντεμαχίων10 διαφορετικών μετοχών.

Να γράψετε αλγόριθμο σε Python οοποίος:

α. Για καθεμία από τις 10 μετοχές διαβάζει

- το όνομα της μετοχής,
- το πλήθος των τεμαχίων της μετοχής, που κατέχει οεπενδυτής, ελέγχοντας το πλήθος να είναι θετικός αριθμός, και καταχωρίζει τα δεδομένα αυτά σε σχετικέςλίστες Α και Β.

β. Για καθεμία από τις 10 μετοχές και για κάθε μία από τις πέντε(5) εργάσιμες ημέρες της εβδομάδας διαβάζει τηντιμή ενός τεμαχίου της μετοχής και την αποθηκεύει σεκατάλληληλίσταΜ, ελέγχοντας η τιμήτου τεμαχίου να είναι θετικός αριθμός.

γ. Για καθεμία από τις 10 μετοχές υπολογίζει τη μέσηεβδομαδιαία τιμή του τεμαχίου της και την αποθηκεύεισε λίστα Κ.

δ. Υπολογίζει και εμφανίζει τη συνολική αξία όλων τωντεμαχίων όλων των μετοχών του επενδυτή, την τελευταίαημέρα της εβδομάδας.

ε. Υπολογίζει εάν ο επενδυτής στο τέλος της εβδομάδας έχεικέρδος ή ζημία ή καμία μεταβολή σε σχέση με το αρχικόποσό που διέθεσε, εμφανίζοντας κατάλληλα μηνύματα.

ΑΣΚΗΣΗ 1

```
def   bubbleSort(A,B):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]>A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
ON=[]
P=[]

for x in range(20):
   on=str(raw_input("Δώσε ονομα"))
   ON.append(on)
   p=input("Δώσε ψήφους")
   P.append(p)

bubbleSort(P,ON)

for x in range(7):
   print ON[x]

onoma=raw_input("Δώσε ονομα")

if onoma in ON[ :7]:
   print "εκλέγεται "
else:
   print "δεν εκλέγεται "
```

ΑΣΚΗΣΗ 2

```
A=[]
B=[]
M=[]
K=[]

for x in range(10):
   on=raw_input("Δώσε ονομα μετοχής")

t=input("Δώσε πλήθος τεμαχίων μετοχής")
whilet<=0:
t=input("Δώσε ΣΩΣΤΟ πλήθος τεμαχίων μετοχής")

A.append(on)
   B.append(T)

   T=[]
   for y in range(5):
      print "Ημερα ",y+1
b=input("Δώσε τιμή τεμαχίου")
T.append(b)
   M.append(b)

S1=0
for x in range(10):
   S=0.0
   for y in range(5):
      S=S+M[x][y]

   K.append(S)

   if y==5:
      S1=S1 + M[x][y] * B[x]

print "αξία τεμαχίων την τελευταία ημέρα της εβδομάδας",S1

if S1>10000:
   print "κέρδος "
elif S1<10000:
   print "ζημία "
else:
   print "καμία μεταβολή "
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2008

1)Μία Νομαρχία διοργάνωσε το2008 σεμινάριο εθελοντικήςδασοπυρόσβεσης, το οποίο παρακολούθησαν500 άτομα.

Η Πυροσβεστική Υπηρεσία ζήτησε στοιχεία σχετικά με τηνηλικία, το φύλο και το μορφωτικό επίπεδο εκπαίδευσηςκάθε εθελοντή, προκειμένου να εξαγάγει στατιστικάστοιχεία.

Να γραφεί αλγόριθμος σε Python, ο οποίος:

α. διαβάζει για κάθε άτομο

• το ονοματεπώνυμο,

• το έτος γέννησης (χωρίς να απαιτείται έλεγχος εγκυρότητας),

• το φύλο, με αποδεκτές τιμές το“Α” για τους άνδρες και το“Γ” για τις γυναίκες,

• το μορφωτικό επίπεδο εκπαίδευσης, με αποδεκτές τιμές “Π”, “Δ” ή “Τ”, που αντιστοιχούν σε Πρωτοβάθμια, Δευτεροβάθμια ή Τριτοβάθμια Εκπαίδευση, και τα καταχωρίζει σε κατάλληλες λίστες.

β. υπολογίζει και εμφανίζει το πλήθος των ατόμων μεηλικία μικρότερη των30 ετών.

γ. υπολογίζει και εμφανίζει το ποσοστό των γυναικών μεεπίπεδο Τριτοβάθμιας Εκπαίδευσης στο σύνολο των εθελοντριών.

δ. εμφανίζει τα ονόματα των ατόμων με τη μεγαλύτερη ηλικία.

2) Σε ένα Δήμο υπάρχουν4 σταθμοί μέτρησης ενός συγκεκριμένουατμοσφαιρικού ρύπου. Η καταγραφή της τιμής του ρύπου γίνεταιανά ώρα και σε 24ωρη βάση. Οι αποδεκτές τιμές του ρύπουκυμαίνονται από 0 έως και100.

Να γραφεί αλγόριθμος σε Python, οοποίος:

α. για κάθε σταθμό και για κάθε ώρα του 24ώρου διαβάζειτην τιμή του ρύπου και την καταχωρίζει σε λίστες A, B, C, D, ελέγχοντας την εγκυρότητα κάθετιμής.

β. για κάθε ώρα του 24ώρου υπολογίζει και εμφανίζει τημέση τιμή του ρύπου από τους 4 σταθμούς.

γ. για κάθε σταθμό βρίσκει και εμφανίζει τη μέγιστη τιμή του ρύπου στο24ωρο.

δ. βρίσκει και εμφανίζει τη μέγιστη τιμή του ρύπου στη διάρκεια του 24ώρου, καθώς και την ώρα και τον αριθμό του σταθμού που σημειώθηκε η τιμή αυτή. (Να θεωρήσετε ότι η τιμή αυτή είναι μοναδική στην λίστα).

ΑΣΚΗΣΗ 1

```
ON=[]
E=[]
F=[]
M=[]


for x in range(500):
   on=str(raw_input("Δώσε ονομα"))
   ON.append(on)

   e=input("Δώσε το έτος")
   E.append(e)

   f=str(raw_input("Δώσε το φύλο"))
F.append(f)

m=str(raw_input("Δώσε το μορφωτικό επίπεδο"))
while m not in ["Π","Δ","Τ"]:
      m=str(raw_input("Δώσε Π ή Δ ή Τ"))

c=0
for x in E:
   if 2008-x <30:
      c=c+1
print "ατόμα με ηλικία μικρότερη των 30",c

c=0
for x in range(500):
   if F=="Γ" and M=="Τ":
      c=c+1
print "ποσοστό γυναικών με επίπεδο Τριτοβάθμιας", c*100/500

min1=2009
for x in E:
   if x<min1:
      min1=x

for x in range(500):
   if M[x]==min1:
      print ON[x]
```

ΑΣΚΗΣΗ 2

```
A=[] ;  B=[]
C=[]
D=[]

for x in range(24):
   a=input("Δώσε τιμή σταθμού 1")
while a<1 or a>100:
      a=input("Δώσε ΣΩΣΤΗ τιμή σταθμού 1")

   b=input("Δώσε τιμή σταθμού 2")
whileb<1 orb>100:
b=input("Δώσε ΣΩΣΤΗ τιμή σταθμού 2")

c=input("Δώσε τιμή σταθμού 3")
whilec<1 orc>100:
c=input("Δώσε ΣΩΣΤΗ τιμή σταθμού 3")

d=input("Δώσε τιμή σταθμού 4")
whiled<1 ord>100:
d=input("Δώσε ΣΩΣΤΗ τιμή σταθμού 4")

A.append(a)
   B.append(b)
   C.append(c)
   D.append(d)

for x in range(24):
   print "ΩΡΑ :", x+1
MO= (A[x]+B[x]+C[x]+D[x])/4.0
   print "Μέσος Όρος σταθμων",MO

max1=0
max2=0
max3=0
max4=0

for x in range(24):
   if A[x]>max1:
      max1=A[x]
      T1=x

   if B[x]>max2:
      max2=B[x]
      T2=x

   if C[x]>max3:
      max3=C[x]
      T3=x

   if D[x]>max4:
      max4=D[x]
      T4=x
print "μέγιστη  τιμή σταθμού 1",max1
print "μέγιστη  τιμή σταθμού 2",max2
print "μέγιστη  τιμή σταθμού 3",max3
print "μέγιστη  τιμή σταθμού 4",max4

MaxAll=max1
Tall=T1
Z=1
if max2>MaxAll:
   MaxAll=max2
   Tall=T2
   Z=2
if max3>MaxAll:
   MaxAll=max3
   Tall=T3
   Z=3
if max4>MaxAll:
   MaxAll=max4
   Tall=T4
   Z=4
print "μέγιστη  τιμή",MaxAll,"την ώρα:",Tall,"του σταθμού",Z
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2009

1)Σε μια διαδρομή τρένου υπάρχουν20 σταθμοί(σε αυτούς περιλαμβάνονται η αφετηρία και ο τερματικός σταθμός). Τοτρένο σταματά σε όλους τους σταθμούς. Σε κάθε σταθμόεπιβιβάζονται και αποβιβάζονται επιβάτες. Οι πρώτοιεπιβάτες επιβιβάζονται στην αφετηρία και στον τερματικόσταθμό αποβιβάζονται όλοι οι επιβάτες.

Να κατασκευάσετε αλγόριθμο σε Python, ο οποίος να διαχειρίζεται τηνκίνηση των επιβατών.

Συγκεκριμένα:

Α. Να ζητάει από το χρήστη τον αριθμό των ατόμων πουεπιβιβάστηκαν σε κάθε σταθμό, εκτός από τον τερματικό, και να τον εισάγει σε λίστα ΕΠΙΒ.

Β. Να εισάγει σε λίστα ΑΠΟΒ τον αριθμό των ατόμωνπου αποβιβάστηκαν σε κάθε σταθμό, εκτός από τοντερματικό, ως εξής:

Για την αφετηρία να εισάγει την τιμή μηδέν(0) και γιατους υπόλοιπους σταθμούς να ζητάει από τον χρήστη τοναριθμό των ατόμων που αποβιβάστηκαν.

Γ. Να δημιουργεί λίστα ΑΕ, στην οποία νακαταχωρίζει τον αριθμό των επιβατών που βρίσκονταιστο τρένο, μετά από κάθε αναχώρησή του.

Δ. Να βρίσκει και να εμφανίζει τον σταθμό από τον οποίοτο τρένο αναχωρεί με τον μεγαλύτερο αριθμό επιβατών. (Να θεωρήσετε ότι από κάθε σταθμό το τρένο αναχωρείμε διαφορετικό αριθμό επιβατών).

2) Ξενοδοχειακή επιχείρηση διαθέτει25 δωμάτια. Τα δωμάτιααριθμούνται από το1 μέχρι το25. Ο συνολικός αριθμός τωνυπαλλήλων που απασχολούνται ημερησίως στο ξενοδοχείοεξαρτάται από τα κατειλημμένα δωμάτια και δίνεται απότον παρακάτω πίνακα

Αριθμός κατειλημμένων δωματίων	**Συνολικός αριθμός υπαλλήλων**
από 0 μέχρι και 4	3
από 5 μέχρι και 8	4
από 9 μέχρι και 12	5
πάνω από 12	6

Η ημερήσια χρέωση για κάθε δωμάτιο είναι75€ και τοημερομίσθιο κάθε υπαλλήλου45€.

Α. Να κατασκευάσετε κύριο πρόγραμμα σε Python το οποίο:

1. Να καταχωρεί σε λίστες Η1, Η2, Η3, Η4, Η5, Η6 και Η7 (μία για κάθε ημέρα της εβδομάδας) την κατάσταση κάθε δωματίου, ελέγχοντας την ορθή καταχώριση. Το πρόγραμμανα δέχεται μόνο τους χαρακτήρες«Κ» γιακατειλημμένο, «Δ» για διαθέσιμο αντίστοιχα.

2. Να υπολογίζει το συνολικό κέρδος ή τη συνολικήζημιά κατά τη διάρκεια της εβδομάδας και ναεμφανίζει κατάλληλο μήνυμα. Για το σκοπό αυτόνα καλεί την συνάρτηση ΚΕΡΔΟΣ, πουπεριγράφεται στο ερώτημα Β.

Β. Να αναπτύξετε την συνάρτηση ΚΕΡΔΟΣ, η οποία ναδέχεται τις 7 λίστες των κρατήσεων και έναν αριθμόημέρας(από1 έως7). Η συνάρτηση να υπολογίζεικαι να επιστρέφει το κέρδος της συγκεκριμένης ημέρας. Το κέρδος κάθε ημέρας προκύπτει από τα ημερήσιαέσοδα ενοικιάσεων, αν αφαιρεθούν τα ημερομίσθια τωνυπαλλήλων της συγκεκριμένης ημέρας. Αν τα έσοδα είναιμικρότερα από τα ημερομίσθια, το κέρδος είναι αρνητικό(ζημιά).

ΑΣΚΗΣΗ 1

```
EPIV=[]
APOV=[]

AE=[]
for i in range(19):

e=input("Δώσε αριθμό των ατόμων που επιβιβάστηκαν ")
EPIV.append(e)

EPIV.append(0)

APOV[0]=0
for i in range(19):

a=input("Δώσε αριθμό των ατόμων που αποβιβάστηκαν ")
APOV.append(e)

AE[0]=EPIV[0]

for x in range(1,21,1):

   AE.append(AE[x-1]+EPIV[x]-APOV[x])

max1=0

for x in range(20):
   if AE[x]>max1:
      max1=AE[x]
Z=x

print "σταθμός με τον μεγαλύτερο αριθμό επιβατών", Z+1
```

ΑΣΚΗΣΗ 2

```
def KERDOS(H1,H2,H3,H4,H5,H6,H7,H):

   if H==1:
      c=0
      for x in range(25):
         if H1[x]=="Κ":
            c=c+1
   if H==2:
      c=0
      for x in range(25):
         if H2[x]=="Κ":
            c=c+1
   if H==3:
      c=0
      for x in range(25):
         if H3[x]=="Κ":
            c=c+1
   if H==4:
      c=0
      for x in range(25):
         if H4[x]=="Κ":
            c=c+1
   if H==5:
      c=0
      for x in range(25):
         if H5[x]=="Κ":
            c=c+1
   if H==6:
      c=0
      for x in range(25):
         if H6[x]=="Κ":
            c=c+1
   if H==7:
      c=0
      for x in range(25):
         if H7[x]=="Κ":
            c=c+1

   if c>=0 and c<=4:
      E=c*75-3*45
   if c>=5 and c<=8:
      E=c*75-4*45
   if c>=9 and c<=12:
      E=c*75-5*45
   if c>12:
      E=c*75-6*45

   return E
```

```
H1=[]
H2=[]
H3=[]
H4=[]
H5=[]
H6=[]
H7=[]

for x in range(1,8,1):
   for y in range(25):
      a=str(raw_input("Δώσεκατάστασηδωματίου"))
      while a not in ["Κ", "Δ"]:
         a=str(raw_input("ΔώσεΚήΔ"))

      if x==1:
         H1.append(a)
      if x==2:
         H2.append(a)
      if x==3:
         H3.append(a)
      if x==4:
         H4.append(a)
      if x==5:
         H5.append(a)
      if x==6:
         H6.append(a)
      if x==7:
         H7.append(a)

AP=KERDOS(H1,H2,H3,H4,H5,H6,H7,H)

if AP>0:
   print "συνολικό κέρδος ",AP, "ευρώ"
if AP<0:
   print "συνολικήζημιά  ",abs(AP), "ευρώ"
if AP==0:
print "Κανένα κέρδος ή ζημιά "
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2009

1)Στις γενικές εξετάσεις, κάθε γραπτό βαθμολογείται από δύοβαθμολογητές στην κλίμακα 1-100. Όταν η διαφορά των δύοβαθμών είναι μεγαλύτερη από δώδεκα μονάδες, το γραπτόαναβαθμολογείται, δηλαδή βαθμολογείται και από τρίτοβαθμολογητή.

Στα γραπτά που δεν έχουν αναβαθμολογηθεί, ο τελικόςβαθμός προκύπτει από το πηλίκο της διαίρεσης τουαθροίσματος των βαθμών των δύο βαθμολογητών διά δέκα.

Στα γραπτά που έχουν αναβαθμολογηθεί, ο τελικός βαθμόςπροκύπτει με τον ίδιο τρόπο, αλλά λαμβάνονται υπόψη οιδύο μεγαλύτεροι βαθμοί.

Για στατιστικούς λόγους, οι τελικοί βαθμοί(ΤΒ) κατανέμονται στις παρακάτω βαθμολογικές κατηγορίες:

1η	**2η**	**3η**	**4η**	**5η**	**6η**
0≤ΤΒ<5	5≤ΤΒ<10	10≤ΤΒ<12	12≤ΤΒ<15	15≤ΤΒ<18	18≤ΤΒ≤20

Σ’ ένα βαθμολογικό κέντρο υπάρχουν780 γραπτά στομάθημα«Ανάπτυξη Εφαρμογών σε ΠρογραμματιστικόΠεριβάλλον». Οι βαθμοί των τριών βαθμολογητών έχουν καταχωριστεί τρείς λίστες Β1, Β2 και Β3.

Να γραφεί πρόγραμμα σε Pythonτο οποίο:

Α. Να ελέγχει, για κάθε γραπτό, αν χρειάζεταιαναβαθμολόγηση. Αν χρειάζεται, να ζητάει από τονχρήστη τον βαθμό του τρίτου βαθμολογητή και να τονεισάγει στην αντίστοιχη θέση της τρίτης στήλης, διαφορετικά να εισάγει την τιμή -1.

Β. Να υπολογίζει τον τελικό βαθμό κάθε γραπτού και νατον καταχωρίζει στην αντίστοιχη θέση μιας λίστας Τ.

Γ. Να εμφανίζει τη βαθμολογική κατηγορία(ή τις κατηγορίες) με το μεγαλύτερο πλήθος γραπτών.

ΑΣΚΗΣΗ 1

```
B1=[]
B2=[]
B3=[]
T=[]

for x in range(780):

   b1=input("Δωσε βαθμο 1ου βαθμολογητη")
whileb1<1 orb1>100:
b1=input("Δωσε ΣΩΣΤΟ βαθμο 1ου βαθμολογητη")
b2=input("Δωσε βαθμο 2ου βαθμολογητη")
whileb2<1 orb2>100:
b2=input("Δωσε ΣΩΣΤΟ βαθμο 2ου βαθμολογητη")

B1.append(b1)
   B2.append(b2)

   ifabs(b1-b2)>12 :
b3=input("Δωσε βαθμο 3ου βαθμολογητη")
whileb3<1 orb3>100:
b3=input("Δωσε ΣΩΣΤΟ βαθμο 3ου βαθμολογητη")
else:
      b3=-1
   B3.append(b3)

for x in range(780):
   if B3[x] ==-1:
      T.append((B1[x]+B2[x])/10)

   else:
      min1=B1[x]
      if B2[x]<min1:
         min1=B1[x]
      if B3[x]<min1:
         min1=B3[x]

      if B1[x]==min1:
         T.append((B2[x]+B3[x])/10)
      if B2[x]==min1:
         T.append((B1[x]+B3[x])/10)
      if B3[x]==min1:
         T.append((B2[x]+B1[x])/10)
```

```
TB=[0,0,0,0,0,0]
NTB=[1,2,3,4,5,6]
for x in range(780):
   if T[x]>=1 and T[x]<5:
      TB[0]=TB[0]+1
   if T[x]>=5 and T[x]<10:
      TB[1]=TB[1]+1
   if T[x]>=10 and T[x]<12:
      TB[2]=TB[2]+1
   if T[x]>=12 and T[x]<15:
      TB[3]=TB[3]+1
   if T[x]>=15 and T[x]<18:
      TB[4]=TB[4]+1
   if T[x]>=18 and T[x]<20:
      TB[5]=TB[5]+1

N=6
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if TB[j]>TB[j-1]:
         TB[j], TB[j-1]= TB[j-1], TB[j]
         NTB[j], NTB[j-1]= NTB[j-1], NTB[j]

for x in range(6):
   if TB[x]==TB[0]:
print "κατηγορία με μεγαλύτερο
πλήθος γραπτών.", NTB[x]
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2009

1) Σε ένα πολυκατάστημα αποφασίστηκε να γίνεται κλιμακωτήέκπτωση στους πελάτες ανάλογα με το ποσό των αγορώντους, με βάση τον παρακάτω πίνακα:

Ποσό αγορών	**Έκπτωση**
έως και300 €	2%
πάνω από300 έως και400 €	5%
πάνω από400 €	7%

Να γραφεί αλγόριθμος σε Pythonπου:

Α. για κάθε πελάτη,

1. να διαβάζει το όνομά του και το ποσό των αγορών του.

2. να υπολογίζει την έκπτωση που δικαιούται.

3. να εμφανίζει το όνομά του και το ποσό που θαπληρώσει μετά την έκπτωση.

Β. να επαναλαμβάνει τη διαδικασία μέχρι να δοθεί ως όνομαπελάτη η λέξη“ΤΕΛΟΣ”.

Γ. να εμφανίζει μετά το τέλος της διαδικασίας τη συνολικήέκπτωση που έγινε για όλους τους πελάτες.

2)Μια επιχείρηση που εμπορεύεται τηλεοράσεις διαθέτει20 μοντέλα.

Να γραφεί αλγόριθμος σε Pythonπου:

Α. να διαβάζει τα ονόματα των μοντέλων και να ταποθηκεύει σε λίστα Α.

Β. να διαβάζει για κάθε μοντέλο τον αριθμό των συσκευώνπου πουλήθηκαν κάθε μήνα, για ένα έτος, και να τοναποθηκεύει σε λίστα Β, ελέγχοντας ώστεο αριθμός αυτός να μην είναι αρνητικός.

Γ. να υπολογίζει και να εμφανίζει το μέσο όρο των ετήσιωνπωλήσεων του κάθε μοντέλου.

Δ. να εμφανίζει κατά αλφαβητική σειρά τα ονόματα τωνμοντέλων καθώς και τον ετήσιο συνολικό αριθμό τωνσυσκευών που πουλήθηκαν για κάθε μοντέλο.

ΑΣΚΗΣΗ 1

```
S=0
on=raw_input("Δώσε όνομα πελάτη")
whileon!="ΤΕΛΟΣ":

p=input("Δώσε ποσό των αγορών")

if p>=1 and p<=300:
      E=P*2/100

   if p>=301 and p<=400:
      E=300*2/100 + (P-300)*5/100

   if p>=401:
      E=300*2/100 + 100*5/100 + (P-
400)*7/100

   print on,E

   S=S+E

   on=raw_input("Δώσε όνομα πελάτη")

print "συνολική έκπτωση ",S
```

ΑΣΚΗΣΗ 2

```
def   bubbleSort(A,B):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]



A=[]
B=[]

for x in range (20):
   on=raw_input("Δώσε όνομα μοντέλου")
   A.append(on)

   T=[]
   for y in range(12):
      print "Μήνας:",y+1
m=input("Δώσε πωλήσεις" )
whilem<0:
m=input("Δώσε ΣΩΣΤΕΣ πωλήσεις" )
T.append(m)
   B.append(T)



for x in range (20):
   S=0.0
   for y in range(12):
      S=S+B[x][y]

print "Μέσος όρος ετήσιων πωλήσεων μοντέλου",x+1,
"είναι",S/12

bubbleSort(A,B)

for x in range (20):
   S=0
   for y in range(12):
      S=S+B[x][y]

print "Σύνολο ετήσιων πωλήσεων μοντέλου",x+1,
"είναι",S
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2010

1)Σε κάποιο σχολικό αγώνα, για το άθλημα«Άλμα εις μήκος» καταγράφεται για κάθε αθλητή η καλύτερη έγκυρη επίδοσήτου. Τιμής ένεκεν, πρώτος αγωνίζεται ο περσινόςπρωταθλητής. Η Επιτροπή του αγώνα διαχειρίζεται ταστοιχεία των αθλητών που αγωνίστηκαν.

Να γράψετε αλγόριθμο σε Python ο οποίος:

Γ1. Να ζητάει το ρεκόρ αγώνων και να το δέχεται, εφόσονείναι θετικό και μικρότερο των10 μέτρων.

Γ2. Να ζητάει τον συνολικό αριθμό των αγωνιζομένων καιγια κάθε αθλητή το όνομα και την επίδοσή του σε μέτραμε τη σειρά που αγωνίστηκε.

Γ3. Να εμφανίζει το όνομα του αθλητή με τη χειρότερηεπίδοση.

Γ4. Να εμφανίζει τα ονόματα των αθλητών που κατέρριψαντο ρεκόρ αγώνων. Αν δεν υπάρχουν τέτοιοι αθλητές, ναεμφανίζει το πλήθος των αθλητών που πλησίασαν τορεκόρ αγώνων σε απόσταση όχι μεγαλύτερη των50 εκατοστών.

Γ5. Να βρίσκει και να εμφανίζει τη θέση που κατέλαβε στηντελική κατάταξη ο περσινός πρωταθλητής.

Σημείωση: Να θεωρήσετε ότι κάθε αθλητής έχει έγκυρηεπίδοση και ότι όλες οι επιδόσεις των αθλητών πουκαταγράφονται είναι διαφορετικές μεταξύ τους.

2)Το ράλλυ Βορείων Σποράδων είναι ένας αγώνας ιστοπλοΐαςανοικτής θάλασσας που γίνεται κάθε χρόνο. Στην τελευταίαδιοργάνωση συμμετείχαν35 σκάφη που διαγωνίστηκαν σε διαδρομήσυνολικής απόστασης70 μιλίων. Κάθε σκάφος ανήκει σε μια από τις κατηγορίεςC1, C2, C3. Επειδή στον αγώνα συμμετέχουν σκάφηδιαφορετικών δυνατοτήτων, η κατάταξη δεν προκύπτει από τον«πραγματικό» χρόνο τερματισμού αλλά από ένα«σχετικό» χρόνο, που υπολογίζεται διαιρώντας τον«πραγματικό» χρόνο του σκάφουςμε τον«ιδανικό». Ο ιδανικός χρόνος είναι διαφορετικός για κάθεσκάφος και προκύπτει πολλαπλασιάζοντας την απόσταση της διαδρομής με τον δείκτηGPH του σκάφους. Ο δείκτηςGPH αντιπροσωπεύει τον ιδανικό χρόνο που χρειάζεται το σκάφος για νακαλύψει απόσταση ενός μιλίου.

Να κατασκευάσετε αλγόριθμο σε Python ο οποίος

Δ1. Να ζητάει για κάθε σκάφος:

- το όνομά του

- την κατηγορία του ελέγχοντας την ορθή καταχώρηση

- τον χρόνο(σε δευτερόλεπτα) που χρειάστηκε για νατερματίσει

- τον δείκτηGPH (σε δευτερόλεπτα).

Δ2. Να υπολογίζει τον σχετικό χρόνο κάθε σκάφους.

Δ3. Να εμφανίζει την κατηγορία στην οποία ανήκουν ταπερισσότερα σκάφη.

Δ4. Να εμφανίζει για κάθε κατηγορία καθώς και για την γενικήκατάταξη τα ονόματα των σκαφών που κερδίζουν μετάλλιο.

(Μετάλλια απονέμονται στους3 πρώτους κάθε κατηγορίας και στους 3 πρώτους της γενικής κατάταξης).

Σημείωση: Να θεωρήσετε ότι κάθε κατηγορία έχει διαφορετικόαριθμό σκαφών και τουλάχιστον τρία σκάφη.

ΑΣΚΗΣΗ 1 (Α΄ Τρόπος)

```
ON=[]
E=[]

R=input("Δωσε ρεκόρ αγώνων ")
whileR<0 orR>= 10:
R=input("Δωσε ΣΩΣΤΟ ρεκόρ αγώνων ")

N=input("Δωσε συνολικό αριθμό αγωνιζομένων
")

for x in range(N):
   on=raw_input("Δωσε ονομα")
   e=input("Δώσε επίδοσή ")

   ON.append(on)
   E.append(e)

min1=E[0]
Z=ON[0]
for x in range(1,N):
   if E[x]<min1:
      min1=E[x]
      Z=ON[x]
print "χειρότερη επίδοση έχει ο ",Z

F=False
for x in range(N):
   if E[x]>R:
      print ON[x]
      F=True
c=0
if F==False:
   for x in range(N):
      ifR-E[x]<50:
c=c+1
print "πλησίασαν  το ρεκόρ  αγώνων  ",c

c=0
for x in range(1,N):
   ifE[x]>E[0]:
c=c+1
print "θέση περσινού πρωταθλητή:",c
```

ΑΣΚΗΣΗ 1 (Β΄ Τρόπος)

```
ON=[]
E=[]

R=input("Δωσε ρεκόρ αγώνων ")
whileR<0 orR>= 10:
R=input("Δωσε ΣΩΣΤΟ ρεκόρ αγώνων ")

N=input("Δωσε συνολικό αριθμό αγωνιζομένων ")

for x in range(N):
   on=raw_input("Δωσε ονομα")
   e=input("Δώσε επίδοσή ")

   ON.append(on)
   E.append(e)

T=E[0]

for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if E[j]>E[j-1]:
         E[j], E[j-1]=E[j-1],E[j]
         ON[j], ON[j-1]=ON[j-1],ON[j]

print "χειρότερη επίδοση έχει ο ",on[-1]

F=False
for x in range(N):
   if E[x]>R:
      print ON[x]
      F=True
c=0
if F==False:
   for x in range(N):
      ifR-E[x]<50:
c=c+1
print "πλησίασαν  το ρεκόρ  αγώνων  ",c

for x in range(N):
   ifE[x]==T:
print "θέση περσινού πρωταθλητή:",x+1
```

ΑΣΚΗΣΗ 2

```
ON=[]
K=[]
SXR=[]

for x  in range(35):
   on=raw_input("Δώσε όνομα σκάφους")

k=str(raw_input("Δώσε κατηγορία σκάφους"))
while K not in ["C1", "C2", "C3"]:
      k=str(raw_input("Δώσε C1 ή C2 ή C3"))

xr=input("Δώσε χρόνο τερματισμού")

gph=input("Δώσε δείκτη GPH ")

Sxr=xr/(70*gph)

   ON.append(on)
   K.append(k)
   SXR.append(sxr)

c1=0
c2=0
c3=0
for x  in K:
   if x=="C1":
      c1=c1+1
   if x=="C2":
      c2=c2+1
   if x=="C3":
      c3=c3+1

if c1>c2 and c1>c3:
   print "C1 κατηγορία  με τα περισσότερα σκάφη"

ifc2>c1 andc2>c3:
print "C2 κατηγορία  με τα περισσότερα σκάφη"

ifc3>c2 andc3>c1:
print "C3 κατηγορία  με τα περισσότερα σκάφη"
```

```
N=35

for i in range(1,N,1):

   for j in range(N-1,i-1,-1):

      if  K[j]<K[j-1]:
         K[j],K[j-1]=K[j-1],K[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         SXR[j],SXR[j-1]=SXR[j-1],V[j]

      if  K[j]==K[j-1]  and  SXR[j]<SXR[j-1]:
         K[j],K[j-1]=K[j-1],K[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         SXR[j],SXR[j-1]=SXR[j-1],V[j]


print "Μετάλλια C1 ",ON[0], ON[1], ON[2]

x=3
while K[x] !="C2":
   x=x+1
print "Μετάλλια C2 ",ON[x], ON[x+1], ON[x+2]

x=x+3
while K[x] !="C3":
   x=x+1
print "Μετάλλια C3 ",ON[x], ON[x+1], ON[x+2]


N=35

for i in range(1,N,1):

   for j in range(N-1,i-1,-1):

      if  SXR[j]<SXR[j-1]:
         K[j],K[j-1]=K[j-1],K[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         SXR[j],SXR[j-1]=SXR[j-1],V[j]

print "ΜετάλλιαΓενΚατατ" , ON[0], ON[1], ON[2]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2010

1) Ένα σύστημα υπολογιστή χρησιμοποιεί για τον έλεγχο πρόσβασης των χρηστών του τρείς λίστες 1000 στοιχείων με τα στοιχεία τους. Στην λίστα **ΟΝ** αποθηκεύει το όνομα πρόσβασης του χρήστη, στην λίστα **Ρ** το συνθηματικό του και στην λίστα **Κ**έναν από τους χαρακτήρες «Σ» ή «Α». (Ο χαρακτήρας «Σ» δηλώνει ότι το συνθηματικό συνεχίζει να ισχύει, ενώ ο χαρακτήρας «Α» δηλώνει ότι το συνθηματικό πρέπει να αλλάξει).

Θεωρήστε ότι υπάρχει ένα κύριο πρόγραμμα που υλοποιεί τα παραπάνω και καλεί τη διαδικασία ΕΛΕΓΧΟΣ η οποία ελέγχει την πρόσβαση του χρήστη στο σύστημα.

Να γράψετε τη συνάρτησηΕΛΕΓΧΟΣ η οποία:

Γ1. Διαβάζει το όνομα και το συνθηματικό του χρήστη. Ελέγχει αν το όνομα πρόσβασης και το συνθηματικό είναι έγκυρα, δηλαδή υπάρχουν στις λίστες ΟΝ και Ρ και αναφέρονται στον ίδιο χρήστη. Αν υπάρχουν, εμφανίζει το μήνυμα «ΚΑΛΩΣ ΗΡΘΑΤΕ», διαφορετικά εμφανίζει το μήνυμα «ΛΑΘΟΣ ΟΝΟΜΑ ΠΡΟΣΒΑΣΗΣ Ή ΣΥΝΘΗΜΑΤΙΚΟ» και ζητά εκ νέου την εισαγωγή των δυο αυτών στοιχείων (ονόματος πρόσβασης και συνθηματικού) μέχρι να δοθούν έγκυρα στοιχεία.

Γ2. Μετά την εμφάνιση του μηνύματος «ΚΑΛΩΣ ΗΡΘΑΤΕ» ελέγχει αν το συνθηματικό χρειάζεται αλλαγή. Αν χρειάζεται, ζήτα από τον χρήστη την εισαγωγή νέουσυνθηματικού δυο φορές (η δεύτερη ως επιβεβαίωση) μέχρις ότου το συνθηματικό και η επιβεβαίωσή του ταυτιστούν. Όταν ταυτιστούν, η διαδικασίααντικαθιστά το παλιό συνθηματικό με το νέο και τον αντίστοιχο χαρακτήρα «Α» της τρίτης στήλης με το «Σ».

2)Ερευνητές που ασχολούνται με μοντέλα προσομοίωσης εξάπλωσης επιδημιών χρησιμοποιούν για τις μελέτες τους μία λίστα **Μ**. Κάθε κελί της λίστας αυτής αντιπροσωπεύει ένα άτομο σε μια περιοχή 5000 κατοίκων στην οποία υπάρχουν εστίες μιας συγκεκριμένης μολυσματικής ασθένειας (επιδημίας). Από σύμβαση η τιμή μηδέν 0 σε ένα κελί αντιπροσωπεύει ένα υγιές άτομο, ενώ η τιμή -1 αντιπροσωπεύει ένα άτομο που έχει τη συγκεκριμένη ασθένεια (μολυσμένο άτομο). Κάθε άτομο έρχεται σε επαφή με τα γειτονικά του και η ασθένεια μπορεί να μεταδοθεί από τον ένα στον άλλο. (Γειτονικά χαρακτηρίζονται δυο άτομα, όταν τα κελιά της λίστας που τα αντιπροσωπεύουν έχουν μια κοινή πλευρά).

Θεωρήστε ότι δίνεται η λίστα Μ που περιέχει ήδη έναν αριθμό μολυσμένων ατόμων. Να υλοποιήσετε πρόγραμμα σε Pythonτο οποίο:

Δ1. Υπολογίζει και εμφανίζει με κατάλληλο μήνυμα τον συνολικό αριθμό των μολυσμένων ατόμων που υπάρχουν στο σύνολο του πληθυσμού.

Δ2. Αποθηκεύει σε κάθε κελί της λίσταςΡ που αντιπροσωπεύει ένα υγιές άτομο έναν αριθμό ο οποίος δείχνει με πόσα μολυσμένα άτομα γειτονεύει το υγιές.

Δ3. Βρίσκει αν υπάρχει έστω και μία «σημαντική» εστία μόλυνσης. Αν υπάρχει, εμφανίζει το μήνυμα «Υπάρχει σημαντική εστία μόλυνσης» μαζί με τη θέση του πρώτου κελιού της εστίας. Αν δεν υπάρχει, εμφανίζει το μήνυμα «Δεν υπάρχει σημαντική εστία μόλυνσης». (Μια εστία μόλυνσης χαρακτηρίζεται σημαντική, όταν δυο ή περισσότερα μολυσμένα άτομα βρίσκονται σε συνεχόμενα γειτονικά κελιά).

ΑΣΚΗΣΗ 1

```
def ELENXOS(ON, P, K):
   onoma=raw_input("Δωσε ονομα")
   pw=raw_input("Δώσε συνθηματικό")
   F=False
   while F==False:
      if onoma in ON:
         for x in range(1000):
            if onoma==ON[x] and pw==P[x]:
               print "ΚΑΛΩΣ ΗΡΘΑΤΕ"
               F=True
               if K[x]=="A":
                  pw1=raw_input("Δώσε ΝΕΟ συνθηματικό")
pw2=raw_input("επιβεβαίωσε το συνθηματικό")
whilepw1!=pw2:
pw2=raw_input("επιβεβαίωσε το συνθηματικό")
K[x]="Σ"
ifF==False:
print "ΛΑΘΟΣ ΟΝΟΜΑ ΠΡΟΣΒΑΣΗΣ Ή ΣΥΝΘΗΜΑΤΙΚΟ"
onoma=raw_input("Δωσε ΣΩΣΤΟ ονομα")
pw=raw_input("Δώσε ΣΩΣΤΟ συνθηματικό")

ON=[]
P=[]
K=[]

for x in range(1000):
   on=str(raw_input("Δώσε ονομα"))
   p=str(raw_input("Δώσε συνθηματικό"))
k=str(raw_input("Δώσε κατάσταση συνθηματικού"))
while k not in ["Σ", "A"]:
      k=str(raw_input("Δώσε Σ ή Α"))

ON.append(on)
   P.append(p)
   K.append(k)


ELENXOS(ON, P, K)
```

ΑΣΚΗΣΗ 2

```
P=[]

c=0
for x in range(5000):
   if M[x]==-1:
      c=c+1

   if x==0:
      if M[x]==0 and M[x+1]==-1:
         P[x]=1
   elif x==4999:
      if M[x]==0 and M[x-1]==-1:
         P[x]=1
   else:
      if (M[x]==0 and M[x-1]==-1) or (M[x]==0 and M[x+1]==-1):
         P[x]=1
      if M[x]==0 and M[x-1]==-1 and M[x+1]==-1:
         P[x]=2

print "Σύνολο μολυσμένων ατόμων",c

c=0
for x in range(5000):

   if M[x]==-1:
      c=c+1

   if M[x]==0:
      ifc>2:
print "Υπάρχει σημαντική εστία μόλυνσης", x-c

c=0
```

ΑΠΟΛΥΤΗΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2010

1) Σε ΚΤΕΟ της χώρας το2010 προσέρχονται οχήματα για έλεγχο. Τα οχήματα είναι τριών κατηγοριών ΦΟΡΤΗΓΟ, ΕΠΙΒΑΤΗΓΟ, ΔΙΚΥΚΛΟ και πληρώνουν 60€, 40€ και 20€ αντίστοιχα. Ένα όχημα χαρακτηρίζεται ως προς την προσέλευσή του "ΕΜΠΡΟΘΕΣΜΟ" ή "ΕΚΠΡΟΘΕΣΜΟ". Τα οχήματα που προσέρχονται εκπρόθεσμα επιβαρύνονται με πρόστιμο15,80€.

Να αναπτύξετε πρόγραμμα σεPython το οποίο:

Γ1. Για κάθε όχημα το οποίο προσέρχεται στο ΚΤΕΟ για έλεγχο

α. διαβάζει την κατηγορία του, το έτος της πρώτηςκυκλοφορίας και τον τύπο προσέλευσης χωρίς κανέναέλεγχο εγκυρότητας.

β. υπολογίζει και εμφανίζει, με βάση την κατηγορία τουκαι την εμπρόθεσμη ή εκπρόθεσμη προσέλευσή του, τοποσό πληρωμής.

Η διαδικασία εισαγωγής δεδομένων τερματίζει όταν δοθεί η τιμή"Τ" σαν κατηγορία οχήματος.

Γ2. Εμφανίζει το πλήθος των φορτηγών που προσήλθαν στοΚΤΕΟ.

Γ3. Εμφανίζει την κατηγορία του παλαιότερου οχήματος.

Γ4. Εμφανίζει το συνολικό ποσό προστίμου.

2)Σε μια δημοτική δανειστική βιβλιοθήκη υπάρχουν158 μέληπου δανείζονται βιβλία.

Να γραφεί αλγόριθμος σε Pythonπου:

Δ1.

α. Για κάθε μέλος διαβάζει το επώνυμο και το φύλο του(Α=άνδρας, Γ=γυναίκα) και τα αποθηκεύει στις λίστες Μ και F, αντίστοιχα. Να γίνεται έλεγχος εγκυρότητας εισαγωγής του φύλου.

β. Για κάθε μήνα ενός έτους διαβάζει το πλήθος των βιβλίων που δανείστηκε κάθε μέλος και το αποθηκεύει σε λίστα Β.

Δ2. Για κάθε μέλος υπολογίζει το συνολικό αριθμό των βιβλίων που δανείστηκε στο έτος και το αποθηκεύει σε λίστα SUM.

Δ3.

α. Υπολογίζει το συνολικό αριθμό των βιβλίων πουδανείστηκαν οι άνδρες.

β. Υπολογίζει το συνολικό αριθμό των βιβλίων πουδανείστηκαν οι γυναίκες.

γ. Εμφανίζει κατάλληλο μήνυμα που δείχνει αν οι άνδρες ή οι γυναίκες έχουν δανειστεί τα περισσότερα βιβλία. Σεπερίπτωση ίσων συνολικών αριθμών βιβλίων να εμφανίζει τομήνυμα “ΙΣΟΣ ΑΡΙΘΜΟΣ ΒΙΒΛΙΩΝ”.

Δ4. Να διαβάζει ένα επώνυμο και χρησιμοποιώντας τη σειριακή αναζήτηση, σε περίπτωση που το επώνυμο είναι αποθηκευμένο στην λίστα Μ, να εμφανίζει το σύνολο των βιβλίων που δανείστηκε στη διάρκεια του έτους. Σε περίπτωση που το επώνυμο δεν είναι αποθηκευμένο στην λίστα να εμφανίζει το μήνυμα“ΤΟ ΕΠΩΝΥΜΟ ΑΥΤΟ ΔΕΝ ΥΠΑΡΧΕΙ”.

Σημείωση: Δεν απαιτείται κανένας άλλος έλεγχος εγκυρότητας εισαγωγής. Δεν υπάρχει συνωνυμία επωνύμων.

ΑΣΚΗΣΗ 1

```
minE=2010
cf=0
Sp=0

K=raw_input("Δώσε κατηγορία οχήματος")

whileK!="Τ":

E.input("Δώσε έτος πρώτης κυκλοφορίας")

P=raw_input("Δώσε τύπο προσέλευσης")

ifK=="ΦΟΡΤΗΓΟ" andP=="ΕΜΠΡΟΘΕΣΜΟ":
poso= 60
cf=cf+1
ifK=="ΦΟΡΤΗΓΟ" andP=="ΕΚΠΡΟΘΕΣΜΟ":
poso= 60 + 15.80
      cf=cf+1
      Sp=Sp=15.80

   if K=="ΕΠΙΒΑΤΗΓΟ" and P=="ΕΜΠΡΟΘΕΣΜΟ":
      poso=40
   if K=="ΕΠΙΒΑΤΗΓΟ" and P=="ΕΚΠΡΟΘΕΣΜΟ":
      poso= 40 + 15.80
      Sp=Sp=15.80

   if K=="ΔΙΚΥΚΛΟ" and P=="ΕΜΠΡΟΘΕΣΜΟ":
      poso=20
   if K=="ΔΙΚΥΚΛΟ" and P=="ΕΚΠΡΟΘΕΣΜΟ":
      poso= 20 + 15.80
      Sp=Sp=15.80

   print "ποσό πληρωμής", poso


   if E<minE:
      minE=E
      k1=K


   K=raw_input("Δώσε κατηγορία οχήματος")

print "πλήθος των φορτηγών", cf

print "κατηγορία του παλαιότερου οχήματος",k1

print "συνολικό ποσό προστίμου",Sp
```

ΑΣΚΗΣΗ 2

```
def    binarysearch(EP, eponimo) :
   first = 0
   last = len(EP)-1
   found = False
   while   found== False  and  first <= last
:

      mid = ( first + last ) // 2
      if    EP[ mid ] == eponimo :
         found = True
      elif   EP[ mid ] < eponimo :
         first = mid + 1
      else:
         last = mid-1

   if   found == True:
       return   mid
   else:
      return   -1

M=[]
F=[]
B=[]
SUM=[]

for x in range(158):
   on=raw_input("Δώσε επώνυμο")

   f=raw_input("Δώσε φύλο")
   while f not in ["Α", "Γ"]:
      f=raw_input("Δώσε ΣΩΣΤΟ φύλο")

M.append(on)
   F.append(f)

   T=[]
   for y in range(12):
      print "ΜΗΝΑΣ:",y+1
      b=input("Δώσε πλήθος βιβλίων")
      T.append(b)
   B.append(T)

for x in range(158):
   S=0
   for y in range(12):
      S=S+B[x][y]

   SUM.append(S)
```

```
Sa=0
Sw=0
for x in range(158):
   if F[x]=="Α":
      Sa=Sa+SUM[x]
   if F[x]=="Γ":
      Sw=Sw+SUM[x]

print "Σύνολο βιβλίων που δανείστηκαν οι άνδρες",Sa
print "Σύνολο βιβλίων που δανείστηκαν οι γυναίκες",Sw

ifSa>Sw:
print "Άνδρες περισσότερα βιβλία"
ifSa<Sw:
print "Γυναίκες περισσότερα βιβλία"
ifSa==Sw:
print "ΙΣΟΣ ΑΡΙΘΜΟΣ ΒΙΒΛΙΩΝ"


eponimo=raw_input("Δώσε επώνυμο")

H=binarysearch(M, eponimo)

if H!=-1:
print"σύνολο βιβλίων στη διάρκεια  έτους", SUM[H]
else:
print  "ΤΟ ΕΠΩΝΥΜΟ ΑΥΤΟ ΔΕΝ ΥΠΑΡΧΕΙ"
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2011

1)Στις εξετάσεις του ΑΣΕΠ οι υποψήφιοι εξετάζονται σε τρειςθεματικές ενότητες. Ο βαθμός κάθε θεματικής ενότητας είναιαπό1 έως100. Η συνολική βαθμολογία κάθε υποψηφίουπροκύπτει από τον μέσο όρο των βαθμών του στις τρειςθεματικές ενότητες. Ο υποψήφιος θεωρείται ως επιτυχών, ανη συνολική βαθμολογία του είναι τουλάχιστον55 και οβαθμός του σε κάθε θεματική ενότητα είναι τουλάχιστον50.

Να γράψετε αλγόριθμο σε Python ο οποίος:

Για κάθε υποψήφιο:

Γ1. Να διαβάζει το όνομά του και τους βαθμούς του σεκαθεμία από τις τρεις θεματικές ενότητες. (Δεναπαιτείται έλεγχος εγκυρότητας δεδομένων).

Γ2. Να εμφανίζει τον μεγαλύτερο από τους βαθμούςπου πήρε στις τρεις θεματικές ενότητες.

Γ3. Να εμφανίζει το όνομα και τη συνολικήβαθμολογία του στην περίπτωση που είναιεπιτυχών.

Γ4. Ο αλγόριθμος να τερματίζει όταν δοθεί ως όνομα η λέξη“ΤΕΛΟΣ”.

Γ5. Στο τέλος να εμφανίζει το όνομα του επιτυχόντα με τημικρότερη συνολική βαθμολογία. Θεωρήστε ότι είναιμοναδικός.

2) Στην αρχή της ποδοσφαιρικής περιόδου οι22 παίκτες μιας ομάδαςψηφίζουν για τους 3 αρχηγούςπου θα τους εκπροσωπούν. Οι παίκτες είναι αποθηκευμένοι σε λίστα ΟΝ και οι οποίοι αριθμούνται από 0 έως21 δηλαδή από την θέση τους στην λίστα. Κάθε παίκτης μπορεί να ψηφίσει όσουςσυμπαίκτες του θέλει, ακόμα και τον εαυτό του. Τα αποτελέσματατης ψηφοφορίας, των τριών ψήφων κάθε παίκτη καταχωρίζονται σε τρείς λίστες Α, Β και C.

Να γράψετε αλγόριθμο σε Pythono οποίος:

Δ1. Να διαβάζει για κάθε παίκτη τις 3 ψήφους του (ονόματα παιχτών) και να καταχωρεί στις λίστες Α, Β και Cτον αντίστοιχο κωδικό από την λίστα ΟΝ (θέση του παίχτη στην λίστα ΟΝ).

Για κάθε ψήφο να γίνεται έλεγχος ορθότητας τιμών ώστε να υπάρχει στην λίστα ΟΝ το όνομα του παίχτη που ψηφίζεται.

Να γίνεται έλεγχος ορθότητας τιμών ώστε ένα παίχτης να μην μπορεί να ψηφιστεί από τον ίδιο παίχτη παραπάνω από μία φορά.

Δ2. Να εμφανίζει το πλήθος των παικτών που δεν ψηφίστηκαν από κανέναν κανέναν.

Δ3. Να εμφανίζει το πλήθος των παικτών που ψήφισαν τον εαυτότους.

Δ4. Να βρίσκει τον παίκτη που έλαβε τις περισσότερεςψήφους και να εμφανίζει το όνομα του και τις ψήφουςπου έλαβε. Θεωρήστε ότι δεν υπάρχουν ισοψηφίες.

ΑΣΚΗΣΗ 1

```
min1=100

on=raw_input("Δώσεόνομα")

while on!= "ΤΕΛΟΣ" :
   V1=input("Δώσεβαθμό 1")
V2=input("Δώσεβαθμό 2")
V3=input("Δώσε βαθμό 3")

max1=V1

if V2 > max1 :
      max1=V2
   if V3 > max1 :
max1=V3
   print "Ο μεγαλύτερος βαθμός ", max1

MO = (V1+V2+V3)/3
   if V1 >= 50 and V2 >= 50 and V3 >= 50 and MO >= 55 :
      print on, MO

      if MO < min1 :
         min1=MO
         M=on

   on=raw_input("Δώσεόνομα")

print M, min1
```

ΑΣΚΗΣΗ 2

```
def find1(S,a):
   for x in range(len(S)):
      if S[x]== a:
         return x

A=[] ; B=[] ;  C=[]
e=0

for x in range(22):
   Y1=raw_input("Δώσε 1 ψήφο")
   while Y1 not in ON:
      Y1=raw_input("ΔώσεΣΩΣΤΗ 1 ψήφο")

    A.append( find1(ON,Y1) )
if x== find1(ON,Y1):
e=e+1

   Y2=raw_input("Δώσε 2 ψήφο")
   while Y2 not in ON or find1(ON,Y2) in A:
      Y2=raw_input("ΔώσεΣΩΣΤΗ 2 ψήφο")

    A.append( find1(ON,Y2) )
if x== find1(ON,Y2):
e=e+1

   Y3=raw_input("Δώσε 3 ψήφο")
   while Y3 not in ON or find1(ON,Y3) in A or find1(ON,Y3) in B:
      Y3=raw_input("ΔώσεΣΩΣΤΗ 3 ψήφο")

    A.append( find1(ON,Y3) )
if x== find1(ON,Y3):
e=e+1
z=0
for x in range(22):
   if x not in A and x not in B and x not in C:
z=z+1

print "πλήθος  των  παικτών  που  δεν  ψηφίστηκαν",z
print "παικτες που ψήφισαν τον εαυτό τους",e

max1=0
for x in range(22):
   for y in range(22):
      if A[y]==x or B[y]==x or  C[y]==x:
         w=w+1
   if w>max1:
      max1=w
onoma=ON[x]
print "έλαβε  τις  περισσότερες ψήφους ",max1, "Ο", onoma
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2011

1) Ένα πρατήριο υγρών καυσίμων διαθέτει έναν τύπο καυσίμου που αποθηκεύεται σε δεξαμενή χωρητικότητας 10.000 λίτρων. Να αναπτύξετε αλγόριθμο ο οποίος:

Γ1. να διαβάζει την ποσότητα (σε λίτρα) του καυσίμου που υπάρχει αρχικά στη δεξαμενή μέχρι να δοθεί έγκυρη τιμή.

Για κάθε όχημα που προσέρχεται στο πρατήριο:

Γ2. να διαβάζει τον τύπο του οχήματος (“Β” για βυτιοφόρο όχημα που προμηθεύει το πρατήριο με καύσιμο και“Ε” για επιβατηγό όχημα που προμηθεύεται καύσιμο από το πρατήριο).

Γ3. Αν το όχημα είναι βυτιοφόρο τότε να γεμίζει τη δεξαμενή μέχρι την πλήρωσή της.

Αν το όχημα είναι επιβατηγό τότε να διαβάζει την ποσότητα καυσίμου την οποία θέλει να προμηθευτεί και, αν υπάρχει επάρκεια καυσίμου στη δεξαμενή, τότε το επιβατηγό όχημα να εφοδιάζεται με τη ζητούμενη ποσότητα καυσίμου, διαφορετικά το όχημα να μην εξυπηρετείται

Γ4. Η επαναληπτική διαδικασία να τερματίζεται, όταν αδειάσει η δεξαμενή του πρατηρίου ή όταν δεν εξυπηρετηθούν τρία διαδοχικά επιβατηγά οχήματα.

Γ5. Στο τέλος ο αλγόριθμος να εμφανίζει:

α. τη μέση ποσότητα καυσίμου ανά επιβατηγό όχημα που εξυπηρετήθηκε

β. τη συνολική ποσότητα καυσίμου με την οποία τα βυτιοφόρα ανεφοδίασαν τη δεξαμενή.

Σημειώσεις: Δεν απαιτείται έλεγχος εγκυρότητας για τον τύπο του οχήματος. Θεωρήστε ότι στο πρατήριο προσέρχεται ένα τουλάχιστον επιβατηγό όχημα για το οποίο η ποσότητα καυσίμου στη δεξαμενή επαρκεί.

2) Ένας όμιλος αποτελείται από20 εταιρίες. Να γράψετε πρόγραμμα σε Pythonτο οποίο:

Δ1. να διαβάζει τα ονόματα των εταιριών του ομίλου και τα κέρδη τους για κάθε ένα από τα έτη2001 έως και 2005. (Θεωρήστε ότι τα κέρδη είναι θετικοί αριθμοί.)

Δ2. για κάθε εταιρία του ομίλου να καλεί συνάρτηση για τον υπολογισμό του συνολικού κέρδους της εταιρίας στην πενταετία. Στη συνέχεια να υπολογίζει και να εμφανίζει το μέσο ετήσιο κέρδος του ομίλου.

Δ3. για κάθε εταιρία να βρίσκει την τριετία με το μεγαλύτερο συνολικό κέρδος και να εμφανίζει το όνομα της εταιρίας και το πρώτο έτος της συγκεκριμένης τριετίας. (Θεωρήστε ότι η τριετία αυτή είναι μοναδική.)

Δ4. Να κατασκευάσετε τη συνάρτηση που θα χρησιμοποιήσετε στο ερώτημα Δ2.

ΑΣΚΗΣΗ 1

```
D=input("Δώσε αρχική ποσότητα δεξαμενής")
whileD>10000 :
D=input("Δώσε ΣΩΣΤΗ ποσότητα δεξαμενής")

Y=10000-D
c=0
E=0
S=0.0
SB=0
while Y> 0 and c!=3:
car=raw_input("Δώσε τύπο οχήματος ")

if car=="B":
      SB=SB+Y
      Y=0

ifcar=="E" :
p=input("Δώσε ζητούμενη  ποσότητα  καυσίμου")
if p<=Y:
         Y=Y+p
         c=0
         S=S+p
         E=E+1
      else:
         c=c+1

MO=S/E
print "Μέση ποσότητα ανά όχημα που εξυπηρετήθηκε",MO

print "Συνολική ποσότητα όπου βυτιοφόρα ανεφοδίασαν ", SB
```

ΑΣΚΗΣΗ 2

```
def info(E1,E2,E3,E4,E5):

   S=0.0
   for x in range(20):
      SE=0
      SE=E1[x]+E2[x]+E3[x]+E4[x]+E5[x]
      print "συνολικό κέρδος της εταιρίας",SE

S=S+SE
MO=S/20
print "μέσο ετήσιο κέρδος του ομίλου",S

E1=[]
E2=[]
E3=[]
E4=[]
E5=[]

for x in range(20):
   e1=input("Δώσε κέρδη 2001")
   E1.append(e1)
   e2=input("Δώσε κέρδη 2002")
   E2.append(e2)
   e3=input("Δώσε κέρδη 2003")
   E3.append(e3)
   e4=input("Δώσε κέρδη 2004")
   E4.append(e4)
   e5=input("Δώσε κέρδη 2005")
   E5.append(e5)

info(E1,E2,E3,E4,E5)


for x in range(20):
   max1=E1[x]+E2[x]+E3[x]
   z="2001"
   if E2[x]+E3[x]+E4[x] > max1:
      max1=E2[x]+E3[x]+E4[x]
      z="2002"
   if E3[x]+E4[x]+E5[x] > max1:
      max1=E3[x]+E4[x]+E5[x]
      z="2003"

   print "τριετία με το μεγαλύτερο συνολικό κέρδος  ", z
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2011

1) Ένα εμπορικό κατάστημα έχει καταγράψει τις μηνιαίεςεισπράξεις του για τα έτη2009 και2010.

Να γράψετε αλγόριθμο σε Pythono οποίος:

Γ1. Να διαβάζει τις μηνιαίες εισπράξεις για καθένα από ταδύο έτη και να τις καταχωρίζει σε αντίστοιχες λίστες Ε1 και Ε2.

Γ2. Να υπολογίζει και να εμφανίζει τη μεγαλύτερη μηνιαίαείσπραξη για κάθε έτος. Θεωρήστε ότι για κάθε έτος ητιμή αυτή είναι μοναδική.

Γ3. Να εμφανίζει κατάλληλο μήνυμα στην περίπτωση που ομήνας κατά τον οποίο σημειώθηκε η μεγαλύτερη μηνιαίαείσπραξη ήταν ο ίδιος και για τα δύο έτη.

Γ4. Να εμφανίζει τον μέσο όρο των μηνιαίων εισπράξεωνγια κάθε έτος.

Γ5. Να υπολογίζει και να εμφανίζει το πλήθος των μηνώντου έτους2009 κατά τους οποίους η μηνιαία είσπραξηήταν μεγαλύτερη από αυτή του αντίστοιχου μήνα τουέτους2010.

ΑΣΚΗΣΗ 1

```
E1=[]
E2=[]

for x in range(12):
   print "Έτος 2009"
e1=input("Δώσε μηνιαίες εισπράξεις ")

print "Έτος 2010"
e2=input("Δώσε μηνιαίες εισπράξεις ")

E1.append(e1)
   E2.append(e2)

max1=0
max2=0

for x in range(12):
   if E1[x]>max1:
      max1=E1[x]
      Z1=x

   if E2[x]>max2:
      max2=E2[x]
      Z2=x

print "μεγαλύτερη μηνιαία είσπραξη για το 2009",max1

print "μεγαλύτερη μηνιαία είσπραξη για το 2010",max2

ifZ1==Z2:
print "μεγαλύτερη μηνιαία είσπραξη τον ίδιος μήνα"

S1=0.0
S2=0.0
for x in range(12):
   S1=S1+E1[x]

S2=S2+E1[x]

print "μέσος όρος για το έτος 2009",S1/12

print "μέσος όρος για το έτος 2010",S2/12

c=0
for x in range(12):
   ifE1[x]>E2[x]:
c=c+1

print "πλήθος των μηνών 2009 περισσότερες εισπράξεις απο 2010",c
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2012

1) Δημόσιος οργανισμός διαθέτει ένα συγκεκριμένο ποσό για την επιδότηση επενδυτικών έργων. Η επιδότηση γίνεται κατόπιν αξιολόγησης και αφορά δυο συγκεκριμένες κατηγορίες έργων με βάση τον προϋπολογισμό τους. Οι κατηγορίες και τα αντίστοιχα ποσοστά επιδότησης επί του προϋπολογισμού φαίνονται στον παρακάτω πίνακα.

Κατηγορία έργου	**Προϋπολογισμός έργου σε ευρώ**	**Ποσοστό Επιδότησης**
Μικρή	200.000 - 299.999	60%
Μεγάλη	300.000 - 399.999	70%

Η εκταμίευση των επιδοτήσεων των αξιολογηθέντων έργων γίνεται με βάση τη χρονική σειρά υποβολής τους. Μετά από κάθε εκταμίευση μειώνεται το ποσό που διαθέτει ο οργανισμός. Να αναπτύξετε αλγόριθμο σε Pythono οποίος:

Γ1. Να διαβάζει το ποσό που διαθέτει ο οργανισμός για το πρόγραμμα επενδύσεων συνολικά, ελέγχοντας ότι το ποσό είναι μεγαλύτερο από 5.000.000 ευρώ.

Γ2. Να διαβάζει το όνομα κάθε έργου. Η σειρά ανάγνωσης είναι η σειρά υποβολής των έργων. Η επαναληπτική διαδικασία να τερματίζεται, όταν αντί για όνομα έργου δοθεί η λέξη «ΤΕΛΟΣ», ή όταν το διαθέσιμο ποσό έχει μειωθεί τόσο, ώστε να μην είναι δυνατή η επιδότηση ούτε ενός έργου μικρής κατηγορίας. Για κάθε έργο, αφού διαβάσει το όνομά του, να διαβάζει και τον προϋπολογισμό του (δεν απαιτείται έλεγχος εγκυρότητας του προϋπολογισμού).

Γ3. Για κάθε έργο να ελέγχει αν το διαθέσιμο ποσό καλύπτει την επιδότηση, και μόνον τότε να γίνεται η εκταμίευση του ποσού. Στη συνέχεια, να εμφανίζει το όνομα του έργου και το ποσό της επιδότησης που δόθηκε.

Γ4. Ναεμφανίζει το πλήθος των έργων που επιδοτήθηκαν από κάθε κατηγορία καθώς και τη συνολική επιδότηση που δόθηκε σε κάθε κατηγορία.

Γ5. Μετά το τέλος της επαναληπτικής διαδικασίας να εμφανίζει το ποσό που δεν έχει διατεθεί, μόνο αν είναι μεγαλύτερο του μηδενός.

2) Μια εταιρεία ασχολείται με εγκαταστάσεις φωτοβολταϊκών συστημάτων, με τα οποία οι πελάτες της έχουν τη δυνατότητα αφενός να παράγουν ηλεκτρική ενέργεια για να καλύπτουν τις ανάγκες της οικίας τους, αφετέρου να πωλούν την πλεονάζουσα ενέργεια προς 0,55€/kWh, εξασφαλίζοντας επιπλέον έσοδα. Η εταιρεία αποφάσισε να ερευνήσει τις εγκαταστάσεις που πραγματοποίησε την προηγούμενη χρονιά σε δέκα (10) πελάτες που βρίσκονται ο καθένας σε διαφορετική πόλη της Ελλάδας.

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Δ1. α. Να διαβάζει για κάθε πελάτη το όνομά του και το όνομα της πόλης στην οποία διαμένει και να τα καταχωρεί στις λίστες ΟΝ και POL αντίστοιχα.

β. Να διαβάζει το ποσό της ηλεκτρικής ενέργειας σε kWh που παρήγαγαν τα φωτοβολταϊκά συστήματα κάθε πελάτη, καθώς και το ποσό της ηλεκτρικής ενέργειας που κατανάλωσε κάθε πελάτης για κάθε μήνα του έτους (δεν απαιτείται έλεγχος εγκυρότητας των δεδομένων).

Δ2. Να υπολογίζει την ετήσια παραγωγή και κατανάλωση ανά πελάτη καθώς και τα ετήσια έσοδά του σε ευρώ (€) και να τα καταχωρεί στις λίστες Ρ για την παραγωγή και Κ για την κατανάλωση και Ε για τα έσοδα αντίστοιχα. Θεωρήστε ότι για κάθε πελάτη η ετήσια παραγόμενη ηλεκτρική ενέργεια είναι μεγαλύτερη ή ίση της ενέργειας που έχει καταναλώσει.

Δ3. Να εμφανίζει το όνομα της πόλης στην οποία σημειώθηκε η μεγαλύτερη παραγωγή ηλεκτρικού ρεύματος.

Δ4. Να καλεί κατάλληλο υποπρόγραμμα με τη βοήθεια του οποίου θα εμφανίζονται τα ετήσια έσοδα κάθε πελάτη κατά φθίνουσα σειρά. Να κατασκευάσετε το υποπρόγραμμα που χρειάζεται για το σκοπό αυτό.

Δ5. Να εμφανίζει το όνομα του πελάτη με τη μικρότερη παραγωγή ηλεκτρικής ενέργειας και την μεγαλύτερη κατανάλωση. Αν δεν υπάρχει να τυπώνεται κατάλληλο μήνυμα.

ΑΣΚΗΣΗ 1

```
P=input("Δώσε Διαθεσιμο Ποσο")
whileP>5000000:
P=input("Δώσε ΣΩΣΤΟ Διαθεσιμο Ποσο")

Pmax=0
Pmin=0

Smax=0
Smin=0

on=raw_input("Δώσε όνομα έργου")
while on!="ΤΕΛΟΣ" and P>=200000 * 0.6:

   PR=input("Δώσε προϋπολογισμό")

ifPR>=200000 andPR<=299999:
E=PR*60/100
   if PR>=300000 and PR<=399999:
      E=PR*70/100


   if P>=E:
      P=P-E
      print on, E

      if PR>=200000 and PR<=299999:
         Pmin=Pmin+1
         Smin=Smin+E
      if PR>=300000 and PR<=399999:
         Pmax=Pmax+1
         Smax=Smax+E

   on=raw_input("Δώσε όνομα έργου")

print "πλήθος των έργων κατηγορίας Μικρή ",Pmin
print "συνολική επιδότηση κατηγορίας Μικρή ",Smin

print "πλήθος των έργων κατηγορίας Μεγάλη ",Pmax
print "συνολική επιδότηση κατηγορίας Μεγάλη ",Smax

if P>0:
   print "Διαθεσιμο Ποσο",P
```

ΑΣΚΗΣΗ 2

```
def bbs(A, B, C, D, E):
   N=len(A)
   for i in range(1,N,1):
      for j in range (N-1,i-1,-1):
         if E[j]> E[j-1]:
            E[j],E[j-1]=E[j-1],E[j]
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
            C[j],C[j-1]=C[j-1],C[j]
            D[j],D[j-1]=D[j-1],D[j]

   for x  in range(len(E)):
      print A[x], E[x]

ON=[]
POL=[]
P=[]
K=[]
E=[]

for x  in range(10):
   on=str(raw_input('Δώσε όνομα πελάτη'))
   pol=str(raw_input('Δώσε όνομα πόλης'))
   ON.append(on)
   POL.append(pol)
   SK=0
   SP=0
   for y  in range(12):
      p=input("Δώσε παραγωγή")
SP=SP+p
k=input('Δώσε κατανάλωση')
SK=SK+k
Esoda=(SP-SK)*0.55
   P.append(SP)
   K.append(SK)
   E.append(Esoda)

max1=0
for x  in range(10):
   if P[x] > max1:
      max1=P[x]
      Poli=POL[x]

print "Πόλη με μεγαλύτερη παραγωγή ρεύματος", Poli

bbs(ON, POL, P, K, E)
```

```
min1=P[0]
for x  in range(1,10):
   if P[x] < min1:
      min1=P[x]
      Z1=x

max1=0
for x  in range(10):
   if K[x] > max1:
      max1=K[x]
      Z2=x

if Z1==Z2:
   print ON[Z1]

else:
   print "Δεν υπάρχει"
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2012

1)Η κρυπτογράφηση χρησιμοποιείται για την προστασία των μεταδιδομένων πληροφοριών. Ένας απλός αλγόριθμος κρυπτογράφησης χρησιμοποιεί την αντιστοίχιση κάθε γράμματος ενός κειμένου σε ένα άλλο γράμμα της αλφαβήτου.

Για το σκοπό αυτό δύο λίστες Α και Βστις οποίεςΑ περιέχει σε αλφαβητική σειρά τους χαρακτήρες από το Α έως και το Ω. Στη Β βρίσκονται οι ίδιοι χαρακτήρες, αλλά με διαφορετική σειρά. Κάθε χαρακτήρας της Α λίστας κρυπτογραφείται στον αντίστοιχο χαρακτήρα της Β λίστας, που βρίσκεται στην ίδια θέση.

Επίσης, δίνεται λίστα ΚΕΙΜ[500], η οποία περιέχει αποθηκευμένο με κεφαλαία ελληνικά γράμματα το προς κρυπτογράφηση κείμενο. Κάθε χαρακτήρας του κειμένου βρίσκεται σε μίαθέση της λίστας ΚΕΙΜ[500]. Οι λέξεις του κειμένου χωρίζονται με έναν χαρακτήρα κενό (΄ ΄), ενώ στο τέλος του κειμένου μπορεί να υπάρχουν χαρακτήρες κενό (΄ ΄), μέχρι να συμπληρωθεί η λίστα.

Να αναπτύξετε πρόγραμμα σε Pythonτο οποίο:

Γ1. Να εμφανίζει το πλήθος των χαρακτήρων κενό(΄ ΄), που υπάρχουν μετά το τέλος του κειμένου στην λίστα ΚΕΙΜ[500]. Αν δεν υπάρχει χαρακτήρας κενό μετά τον τελευταίο χαρακτήρα του μη κρυπτογραφημένου κειμένου, τότε να εμφανίζεται το μήνυμα: «Το μήκος του κειμένου είναι500 χαρακτήρες». Θεωρήστε ότι η λίστα ΚΕΙΜ[500] περιέχει τουλάχιστον μία λέξη.

Γ2. Να κρυπτογραφεί τους χαρακτήρες της λίστας ΚΕΙΜ[500] στην λίστα ΚΡΥΠ[500], με βάση τις λίστες Α και Β. Η κρυπτογράφηση να τερματίζεται με το τέλος του κειμένου. Δίνεται ότι κάθε χαρακτήρας κενό, που υπάρχει στην λίστα ΚΕΙΜ[500], παραμένει χαρακτήρας κενό στην λίστα ΚΡΥΠ[500].

Γ3. Να εμφανίζει το πλήθος των λέξεων του κειμένου, καθώς και το πλήθος των χαρακτήρων που έχει η μεγαλύτερη λέξη του κειμένου στην λίστα ΚΡΥΠ[500]. Θεωρήστε ότι η μεγαλύτερη λέξη είναι μοναδική.

2) Εταιρεία που ασχολείται με μετρήσεις τηλεθέασηςκαταγράφει στοιχεία, ανά ημέρα και για χρονικό διάστημαμίας εβδομάδας, τα οποία αφορούν την τηλεθέαση τωνκεντρικών δελτίων ειδήσεων που προβάλλονται από πέντε(5) τηλεοπτικούς σταθμούς.

Για τη διευκόλυνση της στατιστικής επεξεργασίας τωνδεδομένων να αναπτύξετε πρόγραμμα το οποίο:

Δ1. Για κάθε έναν από τους τηλεοπτικούς σταθμούς ναδέχεται το όνομά του και το πλήθος των τηλεθεατώνπου παρακολούθησαν το κεντρικό δελτίο ειδήσεων κάθεμέρα της εβδομάδας, από Δευτέρα έως και Κυριακή. Ναμη γίνει έλεγχος εγκυρότητας.

Δ2. Να καλεί για κάθε έναν από τους τηλεοπτικούςσταθμούς κατάλληλο υποπρόγραμμα, το οποίο ναυπολογίζει και να επιστρέφει το μέσο πλήθοςτηλεθεατών, που παρακολούθησαν το κεντρικό δελτίοειδήσεών του, τη συγκεκριμένη εβδομάδα.

Να αναπτύξετε το κατάλληλη συνάρτηση.

Δ3. Να εμφανίζει τα ονόματα των σταθμών για τους οποίουςο μέσος όρος τηλεθέασης του Σαββατοκύριακου(2 ημέρες) ήταν μεγαλύτερος από το μέσο όροτηλεθέασης στις καθημερινές(Δευτέρα έως και Παρασκευή).

Δ4. Να εμφανίζει τα ονόματα των τηλεοπτικών σταθμών, οιοποίοι κάθε ημέρα, από Δευτέρα έως και Κυριακή, παρουσιάζουν συνεχώς, από ημέρα σε ημέρα, αύξησητηλεθέασης. Αν δεν υπάρχουν τέτοιοι σταθμοί, ναεμφανίζει το μήνυμα: «Κανένας σταθμός δεν είχε συνεχήαύξηση τηλεθέασης».

ΑΣΚΗΣΗ 1

```
A=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
B=['O','U' ,'H' ,'W' , 'F', 'T','Z' , 'C', 'E', 'J', 'K', 'Y', 'M', 'X', 'A', 'P', 'Q', 'R', 'S', 'I', 'B', 'V', 'D', 'N','L' , 'G']

KEIM=['T', 'H', 'E', ' ', 'P', 'R', 'O', 'B', 'L', 'E', 'M', ' ', 'W', 'I', 'T', 'H', ' ', 'D', 'O', 'I', 'N', 'G', ' ', 'T', 'H', 'A', 'T', ',',
' ', 'I', 'S', ' ', 'T', 'H', 'A', 'T', ' ', 'T', 'O', ' ', 'G', 'E', 'N', 'E', 'R', 'A', 'T', 'E', ' ', ' ']


KRYP=[]

x=0
F=False
while  F==False:
   if KEIM[x]==" " and KEIM[x+1]==" ":
      F=True
   else:
      x=x+1

print "πλήθος χαρακτήρων κενό στο τέλος",len(KEIM)-x


for x in range(len(KEIM)):
   if KEIM[x]==" ":
      KRYP.append(" ")
   else:
      for y in range(len(A)):
         if A[y]== KEIM[x] :
            KRYP.append(B[y])

c=0
for x in range(len(KRYP)-1):
   if KRYP[x]==" " and KRYP[x+1] !=" ":
c=c+1

print "πλήθος  λέξεων  ", c+1

c=0
for x in range(len(KRYP)-1):
   if KRYP[x]==" " and KRYP[x+1]==" ":
      v="TELOS"
   else:
      if KRYP[x]==" ":
         max1=c
         c=0
      else:
         c=c+1

print "μεγαλύτερη λέξη ",c
```

ΑΣΚΗΣΗ 2

```
def MO(B):
   S=0.0
   for x in B:
      S=S+x
   return S/7


A=[]
ON=[]

for x in range(5):
   on=str(raw_input("Δωσε ονομα"))
   ON.append(on)
   B=[]
   for y in range(7):
      p=input("Δωσε αριθμό τηλεθεατών")

B.append(p)
   A.append(B)

   print "μέσο πλήθος τηλεθεατών",MO(B)

S1=0.0
S2=0.0
for x in range(5):
   for y in range(7):
      if y<=5:
         S1=S1+A[x] [y]
      else:
         S2=S2+A[x] [y]

   MO1=S1/5
   MO2=S2/2

   if MO1>MO2:
      print ON[x]

c=0
for x in range(5):
   F=True
   for y in range(6):
      if A[x] [y] > A [x] [y+1]:
         F=Fale
   ifF==True :
print "Σταθμος με αύξηση τηλεθέασης", ON[x]
c=c+1

ifc==0:
print "Κανένας σταθμός δεν είχε συνεχή αύξηση τηλεθέασης"
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2013

1) Σε ένα πρόγραμμα ανταλλαγής μαθητών Comenius συμμετέχουν μαθητές από δυο χώρες: Ελλάδα (EL) και Ισπανία (ES). Οι μαθητές αυτοί καλούνται να απαντήσουν σε μια ερώτηση όπου οι δυνατές απαντήσεις είναι:

1. Πολύ συχνά 2. Συχνά 3. Αρκετές φορές 4. Σπάνια 5. Ποτέ

Στην πρώτη φάση επεξεργασίας της ερώτησης πρέπει να καταγραφούν οι απαντήσεις από κάθε χώρα και να μετρήσουν για κάθε αριθμό απάντησης πόσες φορές υπάρχει, με σκοπό να αναφέρουν για κάθε χώρα, ποια απάντηση είχε τα μεγαλύτερα ποσοστά.

Για να βοηθήσετε στην επεξεργασία να αναπτύξετε πρόγραμμα σε Python το οποίο:

Α. Να δημιουργεί δύο λίστες EL και ES όπου κάθε μια θα έχει 5 θέσεις και αρχικά 0 σε κάθε θέση.

Β. Για κάθε μαθητή να διαβάζει το όνομα της χώρας του και τον αριθμό της απάντησής του. Οι δυνατές τιμές για τη χώρα είναι: EL, ES και για την απάντηση 1,2,3,4,5. Η κάθε απάντηση θα πρέπει να προσμετρείται σε έναν από τις δύο λίστες EL, ES ανάλογα με τη χώρα και στο αντίστοιχο στοιχείο. Δηλαδή, αν δοθούν για τιμές οι ES και 4, τότε θα πρέπει στο 4ο στοιχείο της λίστας ES να προστεθεί μια ακόμα καταχώριση. (Δεν απαιτείται έλεγχος εγκυρότητας τιμών)

Γ. Η προηγούμενη διαδικασία εισαγωγής δεδομένων και καταχώρισης απαντήσεων θα ελέγχεταιαπό την ερώτηση «για Διακοπή της εισαγωγής πατήστε Δ ή δ», που θα εμφανίζεται, και ο χρήστης θα πρέπει να δώσει το χαρακτήρα Δ ή δ για να σταματήσει την επαναληπτική διαδικασία.

Δ. Στο τέλος για κάθε χώρα να εμφανίζει ποιος αριθμός απάντησης είχε το μεγαλύτερο ποσοστό, καθώς και το ποσοστό αυτό. Για την υλοποίηση αυτού του ερωτήματος θα χρησιμοποιήσετε δυο φορές το υποπρόγραμμα ΜΕΓ_ΠΟΣ που θα κατασκευάσετε στο ερώτημα (Ε). Θεωρούμε ότι για κάθε χώρα τα ποσοστά των απαντήσεων είναι διαφορετικά μεταξύ τους και δεν υπάρχει περίπτωση ισοβαθμίας.

Ε. Να αναπτύξετε το υποπρόγραμμα ΜΕΓ_ΠΟΣ το οποίο:

- Να δέχεται μια λίστα ακεραίων 5 θέσεων.
- Να βρίσκει το μεγαλύτερο στοιχείο της λίστας και σε ποια θέση βρίσκεται.
- Να βρίσκει το ποσοστό που κατέχει το μεγαλύτερο στοιχείο σε σχέση με το άθροισμα όλων των στοιχείων της λίστας.
- Να επιστρέφει στο κυρίως πρόγραμμα το ποσοστό αυτό, καθώς και την θέση στην οποία βρίσκεται.

Θεωρήστε ότι όλες οι τιμές των λιστών είναι διαφορετικές και ότι για κάθε χώρα υπάρχει τουλάχιστον μια απάντηση στην ερώτηση.

Θεωρήστε ότι όλες οι τιμές των πινάκων είναι διαφορετικές και ότι για κάθε χώρα υπάρχει τουλάχιστον μια απάντηση στην ερώτηση.

ΑΣΚΗΣΗ 1

```
def MEG_POS(A):
   max1=0
   S=0
   for x in range(5):
      if A[x]>max1:
         max1=A[x]
         Y=x
      S=S+A[x]

   return   x+1,   100*max1/S

EL=[0,0,0,0,0]
ES=[0,0,0,0,0]
E=""

while E!="Δ" and E!="δ":
   XR=raw_input("Δώσε χώρα")
   AR=input("Δώσε απάντηση")

ifXR=="EL":
for x in range(5):
         if AR==x+1:
            EL[x]=EL[x]+1
   if XR=="ES":
      for x in range(5):
         if AR==x+1:
            ES[x]=ES[x]+1

E=raw_input("για Διακοπή της εισαγωγής πατήστε Δ ή δ")

T,P=MEG_POS(EL)

print "Για Ελλάδα περισσότερες απαντήσεις :¨",T
print "Ποσοστό απάντησης :", P

T,P=MEG_POS(ES)

print "Για ΙΣπανία περισσότερες απαντήσεις :¨",T
print "Ποσοστό απάντησης :", P
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2013

1)Ο σύλλογος γονέων και κηδεμόνων μιας περιοχής θέλει να διοργανώσει μιαπολιτιστική εκδήλωση. Για το σκοπό αυτό, ζητά από κάθε σχολείο της περιοχήςνα προσφέρει κάποιο χρηματικό ποσό για την πραγματοποίησή της. Κάθεσχολείο έχει τη δυνατότητα να επικοινωνεί περισσότερες από μία φορές με τοσύλλογο και να τροποποιεί την προσφορά του.

Να αναπτύξετε πρόγραμμα σε Pythonτο οποίο:

Γ1. Να θεωρεί δεδομένημία λίστα Σ[100] που περιέχει τα ονόματα των100 σχολείων της περιοχής και να δημιουργεί λίστα Π[100] που θα περιέχει τις αντίστοιχες χρηματικές προσφορές από κάθε σχολείο. Αρχικά να τοποθετηθεί σε κάθε στοιχείο τις λίστας Π[100] την τιμή -1.

Γ2.

α) Να διαβάζει το όνομα ενός σχολείου και να το αναζητά στην λίστα Σ.

β) Να εμφανίζει το μήνυμα «Άγνωστο», όταν το σχολείο δε βρεθεί. Όταν το σχολείο βρεθεί, να σταματά την αναζήτηση, να διαβάζει τη χρηματική προσφορά του σχολείου και να την τοποθετεί στην αντίστοιχη θέση της λίστας Π. (Όταν δοθεί η τιμή0, σημαίνει ότι το σχολείο δεν μπορεί να προσφέρει χρήματα, δηλαδή έδωσε μηδενική προσφορά). Όταν δεν είναι η πρώτη φορά που δίνει προσφορά τότε να εμφανίζει το μήνυμα «ΤΡΟΠΟΠΟΙΗΣΗ ΠΡΟΣΦΟΡΑΣ» και να αντικαθιστά την προηγούμενη προσφορά του με τη νέα.

Γ3. Να επαναλαμβάνει τις ενέργειες που περιγράφονται στο ερώτημα Γ2, μέχρις ότου όλα τα σχολεία να δώσουν τουλάχιστον μία προσφορά.

Γ4. Να εμφανίζει:

α) το συνολικό χρηματικό ποσό που έχει συγκεντρωθεί,

β) το πλήθος των σχολείων που έδωσαν μηδενική προσφορά,

γ) τοπλήθος των τροποποιήσεων που έγιναν στις προσφορές.

2)Τα δεδομένα (κείμενο, εικόνα, ήχος, κλπ), κατά τη μετάδοσή τους μέσω ενσύρματων ή ασύρματων καναλιών επικοινωνίας, αλλοιώνονται λόγω του θορύβου που χαρακτηρίζει κάθε κανάλι. Ο τρόπος προστασίας των δεδομένων μετάδοσης είναι ο ακόλουθος:

Για κάθε bit (ακέραιος με τιμή 0 ή 1), που ο πομπός θέλει να στείλει, μεταδίδει μια λέξη, που αντιστοιχεί σε λίστα ΜΕΤΑΔΟΣΗ[31] με όλες τις τιμές του ταυτόσημες με το προς μετάδοση bit, δηλαδή, αν πρόκειται να σταλεί το bit1, τότε η λέξη που μεταδίδεται είναι η 11...1 μήκους 31 bits, ενώ αν πρόκειται να σταλεί tobit0, τότε η λέξη που μεταδίδεται είναι η 00...0, μήκους 31 bits. Ο δέκτης λαμβάνει λέξη μήκους 31 bits, τα οποία τοποθετούνται σε λίστα ΛΗΨΗ[31]. Έχουμε «ΛΑΝΘΑΣΜΕΝΗ ΛΗΨΗ», εάν υπάρχει τουλάχιστον ένα στοιχείο της λίστας ΛΗΨΗ[31] με διαφορετική τιμή από αυτήν του αντίστοιχου στοιχείου της λίσταςΜΕΤΑΔΟΣΗ[31]. Εάν το πλήθος των 1 της λίσταςΛΗΨΗ[31] είναι μεγαλύτερο από το πλήθος των 0, τότε ο δέκτης αποφασίζει ότι ο πομπός έστειλε 1, ενώ σε αντίθετη περίπτωση ο δέκτης αποφασίζει ότι ο πομπός έστειλε 0. Σε κάθε περίπτωση, αν περισσότερα από τα μισά των 31 bits της λέξης μετάδοσης έχουν αλλοιωθεί, τότε ο δέκτης θα έχει πάρει «ΛΑΝΘΑΣΜΕΝΗ ΑΠΟΦΑΣΗ».

Να γραφεί πρόγραμμα σε Python, το οποίο να κάνει τα εξής:

Δ1. Για κάθε τιμή ποιότητας του καναλιού, που χαρακτηρίζεται από ακεραίους από 1 έως και 10, να πραγματοποιούνται το πολύ100.000 διαφορετικές προσπάθειες μετάδοσης-λήψης και διόρθωσης λαθών. Εάν όμως ληφθούν 100 λανθασμένες αποφάσεις, τότε να διακόπτεται η διαδικασία για τη συγκεκριμένη τιμή ποιότητας του καναλιού.

Δ2.Σε κάθε προσπάθεια μετάδοσης-λήψης και διόρθωσης λαθών να πραγματοποιούνται οι ακόλουθες ενέργειες:

α. Να διαβάζει (χωρίς έλεγχο εγκυρότητας των τιμών τους) τη μεταδοθείσα λέξη, καθώς και τη ληφθείσα λέξη και να ελέγχει, εάν αυτές ταυτίζονται.

β. Να διορθώνει τη ληφθείσα λέξη στο δέκτη, βάσει της παραπάνω περιγραφής του αλγορίθμου.

Δ3. α. Να αποθηκεύει, για κάθε τιμή ποιότητας καναλιού, σε λίστα ΛΑΘΗΑΠΟΦ[10] το ποσοστό των λανθασμένων αποφάσεων και σε λίστα ΛΑΘΗΛΗΨ[10] το ποσοστό των λανθασμένων λήψεων.

β. Να εμφανίζει συγκεντρωτικά τα ποσοστά των λανθασμένων αποφάσεων και λανθασμένων λήψεων στο δέκτη.

ΑΣΚΗΣΗ 1

```
c=0

S=["SX1", "SX2","SX3","SX4","SX5","SX6","SX7","SX8","SX9", "SX10"]
P=[]

for x in len(S):
   P.append(-1)

while -1 in P:
   on=raw_input("Δώσεόνομα σχολείου")

if on not in S:
      print "Άγνωστο"
   else:
      x=0
      while on !=S[x]:

         if P[x]!=-1:
            print "ΤΡΟΠΟΠΟΙΗΣΗ  ΠΡΟΣΦΟΡΑΣ"
            p=input("Δώσε ΝΕΑ προσφορά")
P[x]=p
c=c+1
else:
p=input("Δωσε προσφορά")
P[x]=p
        x=x+1

SX=0
z=0
for x in P:

   SX=SX+x

   if x==0:
      z=z+1

print "συνολικό  χρηματικό  ποσό  ", SX

print "έδωσαν  μηδενική  προσφορά", z

print "πλήθος των τροποποιήσεων", c
```

ΑΣΚΗΣΗ 2

```
METADOSH=[]
LIPSI=[]
LATHIAPOF=[]
LATHILIPS=[]

for i in range(10):
   N=0                          # σύνολο προσπαθειών
PL=0                           # πλήθος λανθασμένων λήψεων
PA=0                           # πλήθος λανθασμένων αποφάσεων
whilePL< 100 andN< 100000 :
N=N+1
      for j in range(31):
         METADOSH[j]=input("Δωσε τιμη μεταδωσης")
LIPSI[j]=input("Δωσε τιμι λήψης")

M=0
      for j in range(31):
         if METADOSH[j] == LIPSI[j]:
            M=M+1
ifM< 31 :
PL=PL+1
A=0                        # πλήθος 1 της ΛΗΨΗΣ
B=0                        # πλήθος 0 της ΛΗΨΗΣ
      for j in range(31):
         if LIPSI[j] == 1 :
            A=A+1
         else:
            B=B+1
      if A > B :
         G=1                # η μεταβλητή Γ είναι η απόφαση του δέκτη
else:
G=0

ifG != METADOSH[1] :      # αρκεί να μη συμφωνεί με το πρώτο
PA=PA+1

LATHIAPOF.append(PA*100/N )    # δεν μπορεί το N να είναι μηδέν
LATHILIPS.append(PL*100/N)

for i in range(10):
   print LATHIAPOF[i], LATHILIPS[i]
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2013

1)Ανατέθηκε σε μια περιβαλλοντική ομάδα να φτιάξει έναν χάρτη επικινδυνότηταςπυρκαγιών για την οροσειρά του Ταϋγέτου. Ο χάρτης αυτός θα δείχνει σε ποιεςπεριοχές υπάρχει μεγάλη πιθανότητα πυρκαγιάς, σε ποιες μέτρια και σε ποιεςχαμηλή. Για να μπορέσουν να κατασκευάσουν το χάρτη, θα πρέπει σε κάθεπεριοχή να μετρήσουν τη μέση ταχύτητα του αέρα και την υγρασία.

Για ναχαρακτηριστεί μια περιοχή ως **υψηλής επικινδυνότητας** θα πρέπει η μέσηταχύτητα του αέρα να ξεπερνά τα10 m/s και η υγρασία να είναι σε «χαμηλάεπίπεδα».

Για να χαρακτηριστεί ως **μέτριας επικινδυνότητας** θα πρέπει η μέσηταχύτητα του αέρα να ξεπερνά τα10 m/s και η υγρασία να είναι σε «υψηλάεπίπεδα».

Τέλος, για να χαρακτηριστεί ως **χαμηλής επικινδυνότητας** θα πρέπειη μέση ταχύτητα του αέρα να είναι μικρότερη ή ίση των10 m/s ανεξάρτητα απότα επίπεδα της υγρασίας.

Να αναπτύξετε αλγόριθμο σε Python, ο οποίος:

Γ1. Να διαβάζει για10 περιοχές την υγρασία και τη μέση ταχύτητα τουανέμου.

Γ2. Για κάθε περιοχή να εμφανίζει τα μηνύματα«Υψηλή επικινδυνότητα», «Μεσαία επικινδυνότητα» και«Χαμηλή επικινδυνότητα» ανάλογα μετους συνδυασμούς των συνδυασμών μέσης ταχύτητας και υγρασίας.

Γ3. Να εμφανίζει το πλήθος των περιοχών με υψηλή επικινδυνότητα.

ΑΣΚΗΣΗ 1

```
c=0
for x in range(10):

   Y=raw_input("Δώσε υγρασία")

   A=input("Δώσε μέση ταχύτητα ανέμου")

ifA>10 andY=="χαμηλά επίπεδα":
print "υψηλή επικινδυνότητα"
c=c+1

ifA>10 andY=="υψηλά επίπεδα":
print "μέτρια επικινδυνότητα"

ifA<=10 :
print "χαμηλή επικινδυνότητα"

print "πλήθος περιοχών με υψηλή επικινδυνότητα",c
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2014

1) Ένας πελάτης αγοράζει προϊόντα από ένα κατάστημα. Να αναπτύξετε αλγόριθμο σε Pythono οποίος:

Γ1. Για κάθε προϊόν που αγοράζει ο πελάτης, να διαβάζει τον κωδικό του, τον αριθμό τεμαχίων που αγοράστηκαν και την τιμή τεμαχίου. Η διαδικασία ανάγνωσης να σταματά, όταν δοθεί ως κωδικός ο αριθμός 0.

Γ2. Αν ο λογαριασμός δεν υπερβαίνει τα 500 ευρώ, να εμφανίζει το μήνυμα «ΠΛΗΡΩΜΗ ΜΕΤΡΗΤΟΙΣ». Διαφορετικά, να υπολογίζει και να εμφανίζει το πλήθος των απαιτούμενων για την εξόφληση δόσεων, όταν η εξόφληση γίνεται με άτοκες μηνιαίες δόσεις, ως εξής: Τον πρώτο μήνα η δόση θα είναι 20 ευρώ και κάθε επόμενο μήνα θα αυξάνεται κατά 5 ευρώ, μέχρι να εξοφληθεί το συνολικό ποσό.

Γ3. Να υπολογίζει και να εμφανίζει τον συνολικό αριθμό των τεμαχίων με τιμή τεμαχίου μεγαλύτερη των 10 ευρώ.

Γ4. Να υπολογίζει και να εμφανίζει τον συνολικό αριθμό των τεμαχίων με τη μέγιστη τιμή τεμαχίου.

2)Μια εταιρεία Πληροφορικής καταγράφει, για 10 ιστότοπους, τον αριθμό τωνεπισκέψεων που δέχεται ο καθένας, κάθε μέρα, για 360 ημέρες.

Να αναπτύξετε αλγόριθμο σε Python, ο οποίος:

Δ1. Για καθένα από τους ιστότοπους να διαβάζει το όνομά του και τον αριθμό των επισκέψεων που δέχθηκε ο ιστότοπος για καθεμιά ημέρα. Δεν απαιτείται έλεγχος εγκυρότητας τιμών.

Δ2. Να εμφανίζει το όνομα κάθε ιστοτόπου και τον συνολικό αριθμό των επισκέψεων που δέχθηκε αυτός στο διάστημα των 360 ημερών.

Δ3. Να εμφανίζει τα ονόματα των ιστοτόπων που **κάθε μέρα** στο διάστημα των 360 ημερών δέχθηκαν περισσότερες από 500 επισκέψεις. Αν δεν υπάρχουν τέτοιοι ιστότοποι, να εμφανίζει κατάλληλο μήνυμα.

Δ4. Να διαβάζει το όνομα ενός ιστοτόπου. Αν το όνομα αυτό δεν είναι ένα από τα 10 ονόματα που έχουν δοθεί, να το ξαναζητά, μέχρι να δοθεί ένα από αυτά τα ονόματα. Να εμφανίζει τη μέγιστη τιμήεπισκέψεων στον ιστότοπο αυτό.

ΑΣΚΗΣΗ 1

```
S1=0
S=0
A=[]
B=[]
E=[]
code=input(' Δώσεκωδικό ')

while code !=0:
   T=input(' Δώσετεμάχια')
Tm=input(' Δώσε τιμή')
S=S+T*Tm
   A.append(T)
   B.append(Tm)
   E.append(code)
   if T>10:
      S1=S1+Tm
   code=input('Δωσεκωδικό')

if S<=500:
   print "ΠΛΗΡΩΜΗΜΕΤΡΗΤΟΙΣ"
else:
   c=1
   S2=20
   SUM=S
   while SUM>0:
SUM=SUM-5
      c=c+1

   print "Αριθμός δόσεων ", c

print "Αριθμός  τεμαχίων με τιμή τεμαχίου μεγαλύτερη των 10€",S1

max1=A[0]
for x in range(1,len(A)):
   if A[x]>max1:
      max1=A[x]
S3=0
for x in range(len(A)):
   if A[x]==max1:
      S3=S3+B[x]

print "Συνολικό αριθμό τεμαχίων με τη μέγιστη τιμή", S3
```

ΑΣΚΗΣΗ 2

```
ON=[]
A=[]

for x in range(10):
   on=str(raw_input("Δώσε όνομα ιστότοπου"))
   ON.append(on)
   T=[]
   for y in range(360):
      print "Ημέρα ",y+1
e=input("Δώσε αριθμό επισκέψεων ")
T.append(e)
   A.append(T)

for x in range(10):
   S=0
   for y in range(360):
      S=S+A[x] [y]

   print ON[x], S


for x in range(10):
   F=False
   for y in range(360):
      if A[x] [y] <500:
         F=True

   if F==False:
      print "κάθε μέρα περισσότερες από 500 επισκέψεις",ON[x]

onoma=raw_input("Δώσε όνομα ιστότοπου")
while onoma not in ON:

   onoma=raw_input("Δώσε ΣΩΣΤΟ όνομα ιστότοπου")

for x in range (10):
   if ON[x]==onoma:
      Z=x
max1=0
for y in range(360):
   if A[Z] [y] >max1:
      max1= A[Z] [y]

print "μέγιστη τιμή επισκέψεων ", max1
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2014

1)Στις πρόσφατες δημοτικές εκλογές, σε κάποιο δήμο της χώρας, χρησιμοποιήθηκαν για την ψηφοφορία 217 αίθουσες (εκλογικά τμήματα), σε 34 δημόσια κτήρια (εκλογικά καταστήματα). Τα τμήματα αριθμήθηκαν με τη σειρά, από το 1 μέχρι το 217, έτσι ώστε οι αριθμοί των εκλογικών τμημάτων κάθεκαταστήματος να είναι διαδοχικοί: αριθμήθηκαν πρώτα τα τμήματα του πρώτουκαταστήματος, στη συνέχεια τα τμήματα του δεύτερου καταστήματος κ.ο.κ. Τοψηφοδέλτιο ενός από τους συμμετέχοντες συνδυασμούς είχε65 υποψηφίους.

Κάθε ψηφοφόρος ψηφίζει σημειώνοντας σταυρό δίπλα στο όνομα κάθευποψηφίου που επιλέγει.

Να αναπτύξετε πρόγραμμα σε Pythonto οποίο:

Δ1. Να διαβάζει:

α. Το πλήθος των εκλογικών τμημάτων για κάθε εκλογικό κατάστημα.

Να γίνεται έλεγχος εγκυρότητας των τιμών που δίνονται, ώστε αυτέςνα είναι θετικές και το άθροισμά τους να είναι ίσο με217.

β. Τα ονόματα των υποψηφίων του συνδυασμού.

γ. Τον αριθμό των σταυρών που έλαβε καθένας από τους 65 υποψηφίουςτου συνδυασμού, σε κάθε εκλογικό τμήμα.

Δ2. Να εμφανίζει τον συνολικό αριθμό σταυρών που έλαβε κάθε υποψήφιος.

Δ3. Να εμφανίζει τοόνομα του υποψηφίου που έλαβε τους περισσότερουςσυνολικούς σταυρούς στο δεύτερο εκλογικό κατάστημα. (είναι μόνο ένας)

Δ4. Να εμφανίζει, σε αλφαβητική σειρά, τα ονόματα των δέκα πρώτων σεσταυρούς υποψηφίων.

ΑΣΚΗΣΗ 1

```
N=[]
ON=[]  ;  ST=[]  ;  MAXST=[]

while s!=217:
   s = 0
   for x in range(34):
      n=input("Δωσε αριθμό εκλογικών  τμημάτων ")
while n < 0 and s+n >217:
         n=input("Δωσε ΣΩΣΤΟ αριθμό εκλογικών  τμημάτων ")
N.append(n)
      s=s+n

for x in range(65):
   on=raw_input("Δώσε όνομα υποψηφίου ")
   ON.append(on)

for i in range(65):
   A=[]
   for j in range(217):
      st=input("Δώσε αριθμό σταυρών")
A.append(st)
   ST.append(A)

for i in range(65):
   S=0
   for j in range(217):
      S=S+ST[i] [j]
   print "Ο", ON[i], "πήρε ", S, "συνολικά σταυρούς"
MAXST.append(S)

max1=0
for i in range(65):
   S=0
   for y in range(N[0], N[0]+N[1],1):
      S = S+ ST[i] [y]
   if  S > max1:
      max1= S
Z=i

print“έλαβε τους περισσότερουςσυνολικούς σταυρούς”,ON[Z]

N1=65
for i in rnage(1,N1,1):
   for j in range(N1-1,i-1,-1):
      if ON[j]<ON[j-1]:
         ON[j], ON[j-1]=ON[j-1],ON[j]
         MAXST[j], MAXST[j-1]=MAXST[j-1],MAXST[j]
         ST[j], ST[j-1]=ST[j-1],ST[j]

for x in range(10):
   print ON[x]
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2015

1) Μία εταιρεία μεταφοράς δεμάτων διαθέτει δύο αποθήκες Α και Β, στο αεροδρόμιο. Κατά την παραλαβή δεμάτων, κάθε δέμα τοποθετείται στην αποθήκη που έχει εκείνη τη στιγμή τον περισσότερο ελεύθερο χώρο. Αν ο ελεύθερος χώρος της αποθήκης Α είναι ίσος με τον ελεύθερο χώρο της αποθήκης Β, το δέμα τοποθετείται στην αποθήκη Α. Όταν όμως το δέμα δεν χωρά σε καμία από τις δύο αποθήκες, προωθείται στις κεντρικές εγκαταστάσεις της εταιρείας, που βρίσκονται εκτός αεροδρομίου.

[Δ.1.] Να κατασκευάσετε πρόγραμμα που:

α. Να διαβάζει τα μεγέθη ελεύθερου χώρου των αποθηκών Α και Β.

β. Να διαβάζει το μέγεθος κάθε εισερχόμενου δέματος και να εμφανίζει το όνομα της αποθήκης (Α ή Β) στην οποία θα τοποθετηθεί αυτό ή να εμφανίζει το μήνυμα «Προώθηση», όταν το δέμα δεν χωρά σε καμία από τις αποθήκες Α ή Β.

Η διαδικασία παραλαβής τερματίζεται, όταν εισαχθεί ως μέγεθος δέματος η τιμή 0.

γ. Στη συνέχεια, να καλεί υποπρόγραμμα, το οποίο να βρίσκει και να εμφανίζει το όνομα της αποθήκης (Α ή Β) στην οποία τοποθετήθηκαν τα περισσότερα δέματα, ή το μήνυμα «Ισάριθμα» σε περίπτωση που στις δύο αποθήκες Α και Β τοποθετήθηκαν ισάριθμα δέματα, ή το μήνυμα «Καμία αποθήκευση στο αεροδρόμιο», αν κανένα δέμα δεν τοποθετήθηκε σε οποιαδήποτε από τις αποθήκες Α ή Β.

[Δ.2.] Να κατασκευάσετε το υποπρόγραμμα που περιγράφεται στο ερώτημα Δ1.α. β.

2) Ένας διαγωνισμός τραγουδιού διεξάγεται σε δύο φάσεις. Στην πρώτη φάση γίνεται ακρόαση των 45 τραγουδιών που διαγωνίζονται και κάθε μέλος της επταμελούς κριτικής επιτροπής βαθμολογεί το κάθε τραγούδι με βαθμό από 1 έως 10. Στη δεύτερη φάση προκρίνεται κάθε τραγούδι που συγκέντρωσε συνολική βαθμολογία μεγαλύτερη του 50 και το οποίο όλοι οι κριτές έχουν βαθμολογήσει τουλάχιστον με 5. Να γραφεί πρόγραμμα, το οποίο:

Δ1. Για κάθε τραγούδι να διαβάζει τον τίτλο του και τον βαθμό που έδωσε κάθε κριτής. Δεν απαιτείται έλεγχος εγκυρότητας.

Δ2. Να υπολογίζει και να εμφανίζει τη συνολική βαθμολογία του κάθε τραγουδιού, η οποία προκύπτει ως το άθροισμα των βαθμών όλων των κριτών.

Δ3. Να βρίσκει και να εμφανίζει τους τίτλους των τραγουδιών που προκρίνονται στη δεύτερη φάση του διαγωνισμού. Αν κανένα τραγούδι δεν προκρίνεται στη δεύτερη φάση, να εμφανίζει κατάλληλο μήνυμα.

Δ4. Να βρίσκει και να εμφανίζει το πλήθος των κριτών που έδωσαν τον μέγιστο βαθμό τους σε ένα μόνο τραγούδι.

ΑΣΚΗΣΗ 1

```
def  Insert_data():
   global NA
   global NB
   ca=0
   cb=0

   d=input('Dose dema')
   while d!=0:

      if d> NA and d>NB:
         print "Προώθηση"

      else :
         if NA> NB :
            NA=NA-d
            ca=ca+1
         if NB> NA :
            NB=NB-d
            cb=cb+1
         if NA == NB :
            NA=NA-d
            ca=ca+1

      d=input('Dose dema')

   return ca, cb

def  Info(ca, cb):
   if ca>cb:
      print   'A'
   if ca<cb:
      print   'B'
   if ca==cb and ca!=0:
      print   'Ισάριθμα'
   if ca==0 and cb==0:
      print "Καμία αποθήκευση στο αεροδρόμιο"


NA=input('Dose megethos apothikis A')
NB=input('Dose megethos apothikis B')

ca, cb = Insert_data()

Info(ca, cb)
```

ΑΣΚΗΣΗ 2

```
V=[ [ ], [ ], [ ], [ ], [ ], [ ], [] ]

c2=0
for x  in range(45):
   S=0
   F=False
   T=raw_input('Δώσε τίτλο τραγουδιού')
   for y in range (7):
      print "Κριτής No ", y+1
      v=input('Δώσε βαθμό')
      S=S+v

      if v<5:
         F=True

      V[y].append(v)

   print    "Συνολική βαθμολογία τραγουδιού:",  S

   if F==False and S>50:
      print "to", T, "προκρίθηκε"
   else:
c2=c2+1

ifc2==45:
print   "Δεν προκρίθηκε κανένα"


K=0

for x  in range(7):
   for y in range(45):
      if V[x] [y] >max1:
         max1= V[x] [y]
   z=0
   for y in range(45):
      if V[x] [y] == max1:
         z=z+1
   if z==1:
      K=K+1

print     K
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2015

1) Σύμφωνα με το διεθνές σύστημα ονοματολογίας της IUPAC,το όνομα ενός άκυκλου υδρογονάνθρακα $\mathbf{C_xH_y}$ με ευθύγραμμη ανθρακική αλυσίδα αποτελείται από τρία συνθετικά.

Το πρώτο συνθετικό (σ1) καθορίζεται από τον αριθμό χτων ατόμων άνθρακα , ως εξής: Όταν x=1 , η τιμή του σ1 είναι **μεθ·** όταν x=2 , η τιμή του σ1 είναι **αιθ·** όταν x=3 , η τιμή του σ1 είναι **προπ·** όταν x= 4 , η τιμή του σ1 είναι **βουτ ·** όταν x= 5 , η τιμή του σ1 είναι **πεντ ·** όταν x= 6 , η τιμή του σ1 είναι **εξ** κ. ο. κ.

Το δεύτερο συνθετικό (σ2) εξαρτάται από τον αριθμό x των ατόμων του άνθρακα και απ ό τον αριθμό y των ατόμων υδρογόνου και η τιμή του είναι σ2=**ά ν** ή σ2= **έν** ή σ2=**ίν** ή σ2=**αδιεν**, σύμφωνα με τις συνθήκες που φαίνονται στον παρακάτω Πίνακα.

Τιμή του σ2	**Συνθήκη**
άν	y=2x+2 , x≥ 1
έν	y=2x, x≥2
ίν	y=2x - 2, x≥2
αδιέν	y=2x - 2, x≥3

Το τρίτο συνθετικό (σ3) είναι σε κάθε περίπτωση η κατάληξη **ιο** .

Όπως φαίνεται στον Πίνακα, όταν x ≥3 , η τιμή του σ2 είναι ίν ή αδιέν. Ο τρόπος καθορισμού του ορθού ονόματος της ένωσης στην περίπτωση αυτή δεν μας ενδιαφέρει στο πλαίσιο της άσκησης.

Για παράδειγμα, όταν x =3 και y =8, η ένωση είναι το προπ- άν - ιο , ενώ αν x =2 και y =2, η ένωση είναι το προπ- ίν- ιο ή το προπ- αδιέν – ιο, ενώ αν x =3 και y =4, η ένωση είναι το προπ- αδιέν - ιο .

Να κατασκευάσετε πρόγραμμα σε Pythonτο οποίο:

Γ1. Να ζητάει τον αριθμό ατόμων άνθρακα της χημικής ένωσης, κάνοντας έλεγχο εγκυρότητας ώστε αυτός να είναι θετικός.

Γ2 . Να ζητάει τον αριθμό ατόμων υδρογόνου της χημικής ένωσης , κάνοντας έλεγχο εγκυρότητας ώστε να ικανοποιείται τουλάχιστον μία από τις συνθήκες του Πίνακα.

Γ3 . Να εκχωρεί στις μεταβλητές

σ1: το πρώτο συνθετικό του ονόματος της χημικής ένωσης . Θεωρείστεότι δίνεται λίστα Π, σε διαδοχικές θέσεις της οποίας βρίσκονται ήδη καταχωρισμένα τα λεκτικά που αντιστοιχούν στον αριθμό των ατόμων του άνθρακα και

σ3: την κατάληξη του ονόματος της χημικής ένωσης .

Γ4 . Να υπολογίζει το σ2 και να εμφανίζει το όνομα (ή τα ονόματα) της χημικής ένωσης, εμφανίζοντας τα τρία συνθετικά , το ένα δίπλα στο άλλο, χωρισμένα με το χαρακτήρα « – »

2) Μια πολυκατοικία έχει 5 ορόφους, με 8 διαμερίσματα (Δ1, Δ2, …,Δ8) σε κάθε όροφο. Τα διαμερίσματα Δ1 όλων των ορόφων έχουν το ίδιο εμβαδό (Ε1) , τα διαμερίσματα Δ2 όλων των ορόφων έχουν το ίδιο εμβαδό (Ε2) κ.ο.κ. Το ποσό των κοινοχρήστων της πολυκατοικίας κατανέμεται στους 5 ορόφους , σύμφωνα με το ποσοστό συμμετοχής του κάθε ορόφου, όπως φαίνεται στον παρακάτω Πίνακα

Όροφος	**Ποσοστό συμμετοχής**
1	5%
2	15%
3	20%
4	25%
5	35%

Το ποσό των κοινοχρήστων του κάθε ορόφου κατανέμεται στα διαμερίσματα του ορόφου αυτού, ανάλογα με το εμβαδό του καθενός διαμερίσματος.

Να γράψετε πρόγραμμα σε Python, το οποίο :

Δ1 . Να ζητάει :

α. Το συνολικό ποσό κοινοχρήστων της πολυκατοικίας.

β . Τα εμβαδά Ε1, Ε2, … Ε8.

Δ2. Να υπολογίζει το ποσό των κοινοχρήστων που αναλογεί σε κάθε όροφο της πολυκατοικίας .

Δ3. Να υπολογίζει το ποσό των κοινοχρήστων που αναλογεί σε κάθεδιαμέρισμα της πολυκατοικίας .

Δ4. Να εμφανίζει τον αριθμό ορόφου (1 - 5) και τον αριθμό διαμερίσματος (1 - 8) ενός διαμερίσματος στο οποίο αναλογεί ποσό κοινοχρήστων μεγαλύτερο του μέσου όρου όλης της πολυκατοικίας.

ΑΣΚΗΣΗ 1

```
P=["μεθ", "αιθ", "προπ", "βουτ", "πεντ", "εξ"]

X=input("Δώσε αριθμό ατόμων άνθρακα")
whileX<0 :
   X=input("Δώσε ΣΩΣΤΟ αριθμό ατόμων άνθρακα ")

Y=input("Δώσε αριθμό ατόμων υδρογόνου")
whileY<0 orY%2!=0 :
Y=input("Δώσε ΣΩΣΤΟ αριθμό ατόμων υδρογόνου ")

s1=P[X-1]

s3="ιο"

if X >= 1 and Y = 2*X + 2 :
   s2 = "άν"

if X >= 2 and Y = 2*X :
   s2 = "έν"

if X>=2 and Y = 2*X - 2:
   if X>=3:
      s2 = "αδιέν"
      print s1 + "-" + s2 + "-" + s3
   s2 = "ίν"

print s1 + "-" + s2 + "-" + s3
```

ΑΣΚΗΣΗ 2

```
P=input("Δώσε συνολικό ποσό κοινοχρήστων ")

E=[]
OR=[]
for x in range(8):
   print "E", x+1
   e=input("Δώσε εμβαδόν")
   E.append(e)

OR.append(P*5/100)
OR.append(P*15/100 )
OR.append(P*20/100 )
OR.append(P*25/100 )
OR.append(P*35/100 )

for x in range(5):
   print "κοινοχρηστα",x+1," ορόφου:", OR[x]

SE=0
for x in range(8):
   SE=SE+x
S=0.0
for x in range(5):
   S=S+x

MO=S/5

for x in range(5):
   for y in range(8):

      KOR=OR[x]*(100*E[x]/SE)/100     #  OR[x]*E[x]/SE

      print "κοινοχρηστα",x+1," ορόφου και διαμερίσματος E",y+1, "είναι:",KOR

if KOR>MO:
         print "Όροφος ", x+1, "Διαμέρισμα ", y+1
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2016

1) Μία εταιρεία πληροφορικής προσφέρει υπολογιστές σε τιμές οι οποίες μειώνονται ανάλογα με την ποσότητα της παραγγελίας, όπως φαίνεται στον παρακάτω πίνακα:

ΠΟΣΟΤΗΤΑ	**ΤΙΜΗ ΜΟΝΑΔΑΣ**
1-50	580
51-100	520
101-200	470
Πάνω από 200	440

Να κατασκευάσετε πρόγραμμα το οποίο:

Γ 1. Να διαβάζει τον αριθμό υπολογιστών που έχει προς πώληση (απόθεμα), ελέγχοντας ότι δίνεται θετικός αριθμός

Γ2. Για κάθε παραγγελία, να διαβάζει την απαιτούμενη ποσότητα και, εφόσον το απόθεμα επαρκεί για την κάλυψη της ποσότητας να εκτελεί την παραγγελία με την ποσότητα που ζητήθηκε. Αν το απόθεμα δεν επαρκεί, διατίθεται στον πελάτη το διαθέσιμο απόθεμα. Η εισαγωγή παραγγελιών τερματίζεται, όταν εξαντληθεί το απόθεμα.

Για κάθε παραγγελία να εμφανίζει:

Γ 3. το κόστος της παραγγελίας

Γ 4. το επιπλέον ποσό που θα κόστιζε η παραγγελία, εάν ο υπολογισμός γινόταν κλιμακωτά με τις τιμές που φαίνονται στον πίνακα.

2) Το Πανελλήνιο Σχολικό Δίκτυο παρέχει πρόσβαση στο Διαδίκτυο (Ίντερνετ) σε 150.000 μαθητές και διατηρεί τα στοιχεία τους, καθώς και στατιστικά στοιχεία, σχετικά με την πρόσβασή τους στο Διαδίκτυο.

Να κατασκευάσετε πρόγραμμα το οποίο:

Δ1) Για κάθε μαθητή να διαβάζει:

α) τον αλφαριθμητικό κωδικό του και να τον καταχωρίζει σε λίστα με όνομα CODE

β) το φύλο του, «Α» αν είναι αγόρι και «Κ» αν είναι κορίτσι, και να το καταχωρίζει σε λίστα με όνομα F

γ) Να δίνετε από το πληκτρολόγιο τον χρόνο πρόσβασής του στο Διαδίκτυο ανά μήνα, για ένα έτος.

Δ2) Να υπολογίζει και να καταχωρίζει σε λίστα SUM το συνολικό ετήσιο χρόνο πρόσβασης κάθε μαθητή.

Δ3) Να εμφανίζει τον κωδικό του αγοριού με το μεγαλύτερο συνολικό χρόνο πρόσβασης και, στη συνέχεια, τον κωδικό του κοριτσιού με το μεγαλύτερο συνολικό χρόνο πρόσβασης, καλώντας τη συνάρτηση T_MAX, που περιγράφεται στο ερώτημα Δ4, μία φορά για τα αγόρια και μία για τα κορίτσια.

Δ4) Να αναπτύξετε συνάρτηση T_MAX η οποία:

α) να δέχεται ως παραμέτρους: την λίτα του φύλου, την λίστα του συνολικού ετήσιου χρόνου πρόσβασης των μαθητών και τον χαρακτήρα «Α» ή «Κ» που αντιστοιχεί στο φύλο

β) να βρίσκει τη θέση της μέγιστης τιμής του ετήσιου χρόνου πρόσβασης αγοριών ή κοριτσιών, ανάλογα με την τιμή «Α» ή «Κ» του φύλου

γ) να επιστρέφει τη θέση της μέγιστης τιμής

(**Σημείωση**: Δεν απαιτείται έλεγχος εγκυρότητας. Να θεωρήσετε ότι όλες οι εισαγωγές γίνονται σωστά και όλες οι συνολικές τιμές χρόνου πρόσβασης είναι μοναδικές).

ΑΣΚΗΣΗ 1

```
N=input("Δώσε απόθεμα")
whileN<0:
N=input("Δώσε ΣΩΣΤΟ απόθεμα")

P=input('Δώσε παραγγελία')

whileP<=N :

if P>=1 and P<=50:
      K1=P*580
   if P>=1 and P<=100:
      K1=P*520
   if P>=101 and P<=200:
      K1=P*470
   ifP>200:
K1=P*440

print "Κόστος  παραγγελίας", K1

if P>=1 and P<=50:
      K2=P*580
   if P>=1 and P<=100:
      K2=50*580 + (P-50)*520
   if P>=101 and P<=200:
      K2=50*580 + 50*520 + (P-100)*470
   ifP>200:
K2=50*580 + 50*520 + 100*470 + (P-200)*440

print "επιπλέον κόστος",K2-K1

P=input('Δώσε παραγγελία')
   if P>N and N!=0:
      P=N
```

ΑΣΚΗΣΗ 2

```
def T_MAX(A,B,Fi):
   max1=B[0]
   for x in range(1,len(A)):
      if A[x]==Fi :
         if B[x]>max1:
            max1=B[x]
            Z=x
   return Z

CODE=[]
F=[]
SUM=[]

for x in range(150000):
   code=str(raw_input("Δωσε κωδικό"))
   f=str(raw_input("Δωσε φύλο"))
   S=0
   for y in range(12):
      xr=input("Δώσε χρόνο")
      S=S+xr
   CODE.append(code)
   F.append(f)
   SUM.append(S)


Z1=T_MAX(F,SUM,"A")

print "Ο κωδικός για τα αγόρια", CODE[Z1]

Z1=T_MAX(F,SUM,"K")

print "Ο κωδικός για τα κορίτσια", CODE[Z1]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2016

1) Στο πλαίσιο μιας μελέτης, ένας φιλόλογος θέλει να ελέγξει τη χρήση ενός δείγματος εκατό (100) ναυτικών λέξεων σε σύγχρονα νεοελληνικά κείμενα. Για τον σκοπό αυτό:

Γ1. Να κατασκευάσετε υποπρόγραμμα, με όνομα ΑΝΑΖΗΤΗΣΗ, το οποίο να δέχεται για είσοδο

- μία λίστα χαρακτήρων Ρ,
- μια αλφαριθμητική μεταβλητή Χ

Το υποπρόγραμμα να αναζητά μια λέξη, την τιμή της μεταβλητής Χ στην λίστας Ρ. Αν βρεθεί η λέξη , το υποπρόγραμμα να επιστρέφει την θέση που βρέθηκε. Αν δεν βρεθεί, να επιστρέφει την τιμή -1.

Στη συνέχεια να κατασκευάσετε κύριο πρόγραμμα το οποίο :

Γ2. Να ζητά 100 ναυτικές λέξεις και να τις καταχωρίζει σε λίστα ΛΕ. Κάθε λέξη που δίνεται να τη δέχεται, μόνο εφόσον ελέγξει ότι δεν έχει ήδη καταχωριστεί στη λίστα. Ο έλεγχος να γίνεται με τη χρήση του υποπρογράμματος ΑΝΑΖΗΤΗΣΗ.

Γ3. Να ζητά, με τη σειρά, τις λέξεις ενός νεοελληνικού κειμένου. Η εισαγωγή να τερματίζεται όταν δοθεί ως λέξη η ακολουθία χαρακτήρων «ΤΕΛΟΣ_ΚΕΙΜΕΝΟΥ».

Γ4. Να εμφανίζει τις σπανιότερες ναυτικές λέξεις του δείγματος που υπάρχουν στο νεοελληνικό κείμενο, δηλαδή τις λέξεις με τη μικρότερη συχνότητα εμφάνισης , χρησιμοποιώντας κατάλληλα το υποπρόγραμμα ΑΝΑΖΗΤΗΣΗ.

2) Στον αρχαιολογικό χώρο της Πύλου διασώθηκαν θραύσματα κεραμικών πινακίδων στα οποία είχαν καταγραφεί σε γραμμές βασικά αγαθά με τις ποσότητες τους, τα οποία είχε συλλέξει η πόλη κατά τη διάρκεια καλλιεργητικών περιόδων. Σε κάθε θραύσμα, αναφέρονται τα πλήρη στοιχεία (όνομα αγαθού, περίοδος, ποσότητα) για ένα ή περισσότερα αγαθά. Βρέθηκαν στοιχεία για δεκαπέντε (15) βασικά αγαθά και πέντε (5) καλλιεργητικές περιόδους. Όλα τα αγαθά υπάρχουν και στις πέντε περιόδους

Σε κάθε γραμμή οι πρώτοι δέκα χαρακτήρες αντιστοιχούν στο όνομα του αγαθού, ο ενδέκατος στην καλλιεργητική περίοδο και ο δωδέκατος στην ποσότητα που συλλέχτηκε. Οι πέντε καλλιεργητικές περίοδοι αναπαρίστανται από τους χαρακτήρες Α, Β, Γ, Δ και Ε. Η ποσότητα που συλλέχτηκε αναπαρίσταται από τους χαρακτήρες Ι, Κ, Λ , Μ, Ν, Ξ και Ο. Έχει βρεθεί ότι η ποσότητα που αντιστοιχεί σε αυτούς είναι : Ι = 10, Κ = 50 , Λ = 100 , Μ = 500, Ν = 1.000, Ξ = 5.000 και Ο = 10.000.

Συνολικά τα στοιχεία των θραυσμάτων μπορούν να αναπαρασταθούν με λίστα Ρ. Κάθε στοιχείο της λίστας περιέχει τα στοιχεία των αγαθών ως συμβολοσειρά 15 χαρακτήρων (όνομα αγαθού, καλλιεργητική περίοδος, ποσότητα).

Να γράψετε πρόγραμμα σε Python το οποίο:

Δ1. Να εισάγει σε λίστα χαρακτήρων Ρ τα στοιχεία των αγαθών που βρέθηκαν στα θραύσματα των πινακίδων.

Δ2. Να ταξινομεί κατά αύξουσα σειρά την λίστα Ρ, με βάση την καλλιεργητική περίοδο, και για την ίδια καλλιεργητική περίοδο, να ταξινομεί τα αγαθά, με βάση τον πρώτο χαρακτήρα κάθε αγαθού .

(Θεωρήστε ότι ο πρώτος χαρακτήρας κάθε αγαθού είναι μοναδικός).

Δ3.

α. Να δημιουργεί μια λίστα ακεραίων Α[75]. Κάθε στοιχείο της λίστας Α αντιστοιχεί σε ένα στοιχείο της ταξινομημένης λίστας Ρ και περιέχει την αντίστοιχη ποσότητα του αγαθού που συλλέχτηκε. Η μετατροπή της ποσότητας από χαρακτήρα σε αριθμό να γίνει με βάση την αντιστοιχία που δόθηκε παραπάνω.

β. Να βρίσκει και να εμφανίζει για κάθε αγαθό το πρώτο γράμμα του ονόματός του και την καλλιεργητική του περίοδο με τη μέγιστη ποσότητα που συλλέχτηκε. (Θεωρήστε ότι η μέγιστη ποσότητα κάθε αγαθού είναι μοναδική).

Δ4. Να δημιουργεί μια λίστα ακεραίων SUM[15]. Κάθε στοιχείο της λίστας SUM αντιστοιχεί σε ένα αγαθό (όπως αυτό εμφανίζεται στις δεκαπέντε πρώτες σειρές της λίστας Ρ) και περιέχει την συνολική ποσότητα του αγαθού που συλλέχτηκε στις πέντε καλλιεργητικές περιόδους.

ΑΣΚΗΣΗ 1

```
def ANAZHTHSH(P,X):
   for i in range(len(P)):
      if P[i]==X:
          return i
   return -1

LE=[]
TL=[]

for x in range(100):
   le=str(raw_input("Δώσε λέξη"))
   if ANAZHTHSH(LE,le)==-1:
      LE.append(le)

   TL.append(0)

Z=0
K=raw_input("Δώσε λέξεις ενός νεοελληνικού κειμένου")

while K!="ΤΕΛΟΣ_ΚΕΙΜΕΝΟΥ":

   if ANAZHTHSH(LE,K)!=-1:
      TL[ANAZHTHSH(LE,K)]+=1

   Z=Z+1
   K=raw_input("Δώσε λέξεις ενός νεοελληνικού κειμένου")

min1=Z

for x in range(100):
   if TL[x]!=0 and TL[x]<min1:
      min1=TL[x]

print "λέξεις με τη μικρότερη συχνότητα εμφάνισης:"
for x in range(100):
   if TL[x]==min1:
      print LE[x]
```

ΑΣΚΗΣΗ 2

```
P=[] ;  A=[]
SUM=[]

for x in range(75):
   p=str(raw_input("Δώσε τα στοιχεία των αγαθών"))
   P.append(p)

N=75
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):

      Z1=P[j]
      Z2=P[j-1]
      if   Z1[10]<Z2[10]:
         P[j],P[j-1]=P[j-1],P[j]

      if   Z1[10]==Z2[10]   and   Z1[0]<Z2[0]:
         P[j],P[j-1]=P[j-1],P[j]

for X in P:
   if X[-1]=="Ι":
      A.append(10)
   if X[-1]=="Κ":
      A.append(50)
   if X[-1]=="Λ":
      A.append(100)
   if X[-1]=="Μ":
      A.append(500)
   if X[-1]=="Ν":
      A.append(1000)
   if X[-1]=="Ξ":
      A.append(5000)
   if X[-1]=="Ο":
      A.append(10000)

for y in range(15):
   max1=0
   for x in range(0,75,15):
      Z=P[x+y]
      if A[x+y]>max1:
         max1=A[x+y]
         G=Z[0]
         K=Z[10]

   print "πρώτο γράμμα του ονόματός ",G,"καλλιεργητική περίοδος",K

for y in range(15):
   S=0
   for x in range(0,75,15):
      Z=P[x+y]
      S=S+A[x+y]
   SUM.append(S)
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2016

1) Το Εθνικό Αρχαιολογικό Μουσείο, το οποίο γιορτάζει τα 150 χρόνια από τη θεμελίωσή του, θέλει να αναπτύξει μία εφαρμογή για την προβολή των εκθεμάτων του.

Να αναπτύξετε ένα πρόγραμμα σε Python, το οποίο:

Δ1. Να διαβάζει 1.000.000 ακέραιους κωδικούς εκθεμάτων στην λίστα Κ και 1.000.000 ονομασίες εκθεμάτων στην λίστα ΟΝ.

Δ2. Να ταξινομεί, κατά αύξουσα σειρά, τις λίστες με βάση τον κωδικό του εκθέματος.

Δ3. Να ζητά από τον χρήστη την εισαγωγή ενός κωδικού και, εφόσον αυτός αντιστοιχεί σε έκθεμα, να εμφανίζει την ονομασία του εκθέματος. Εάν το έκθεμα δεν υπάρχει, να εμφανίζει το μήνυμα: «Δεν υπάρχει». Η διαδικασία να ολοκληρώνεται, όταν εισαχθεί ως κωδικός ο αριθμός 0.

(Σημείωση: Να θεωρήσετε ότι οι κωδικοί όλων των εκθεμάτων είναιδιαφορετικοί μεταξύ τους).

ΑΣΚΗΣΗ 1

```
def   bubbleSort(A,B):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]

ON=[]
K=[]

for x in range(100000):
   k=input("Δώσε κωδικό εκθέματος")
   on=str(raw_input("Δώσε ονομα εκθέματος"))
   K.append(k)
   ON.append(on)

bubbleSort(K,ON)

code=input("Δώσε κωδικό εκθέματος")

while code!=0:

   if code in K:
      for x in range(100000):
         if K[x]==code:
            print ON[x]
   else:
      print "Δεν  υπάρχει"

   code=input("Δώσε κωδικό εκθέματος")
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2016 - ΠΑΛΑΙΟ ΣΥΣΤΗΜΑ

1) Ένας μαθητής αγόρασε έναν εξωτερικό δίσκο χωρητικότητας 1000 GB, προκειμένου να αποθηκεύσει σε αυτόν ψηφιακά αρχεία.

Να γραφεί πρόγραμμα σε Python, το οποίο:

Γ1. Για κάθε ψηφιακό αρχείο που θέλει να αποθηκεύσει ο μαθητής στον εξωτερικό δίσκο, να διαβάζει το όνομά του και το μέγεθός του (σε GB) και να ελέγχει, αν επαρκεί η διαθέσιμη χωρητικότητα του εξωτερικού δίσκου. Εφόσον επαρκεί, να εμφανίζει το μήνυμα «Επιτρεπτή αποθήκευση» και να υπολογίζει τη νέα διαθέσιμη χωρητικότητα του εξωτερικού δίσκου. Να τερματίζει τον έλεγχο της αποθήκευσης ψηφιακών αρχείων στον εξωτερικό δίσκο, όταν το μέγεθος του αρχείου που θέλει να αποθηκεύσει ο μαθητής είναι μεγαλύτερο από τη διαθέσιμη χωρητικότητα του εξωτερικού δίσκου.

Γ2. Να υπολογίζει και να εμφανίζει το ποσοστό του αριθμού των αρχείων που αποθηκεύτηκαν και έχουν μέγεθος μεγαλύτερο των 10 GB.

Γ3. Να βρίσκει και να εμφανίζει τα ονόματα των δύο μικρότερων σε μέγεθος αρχείων που αποθηκεύτηκαν στον εξωτερικό δίσκο.

Να θεωρήσετε ότι:

α) θα αποθηκευτούν τουλάχιστον δύο αρχεία στον εξωτερικό δίσκο,

β) τα μεγέθη όλων των αρχείων που αποθηκεύονται, είναι διαφορετικά μεταξύ τους.

2) Μια περιβαλλοντική οργάνωση έχει εκπαιδεύσει δέκα (10) εθελοντές οι οποίοι θα ενημερώσουν το κοινό σε θέματα που αφορούν την προστασία του περιβάλλοντος.

Να γράψετε πρόγραμμα σε Python, το οποίο:

Δ1. Για κάθε εθελοντή, να διαβάζει το όνομά του και τον αριθμό των ατόμων που ενημέρωσε κάθε μήνα, στη διάρκεια του προηγούμενου έτους (δεν απαιτείται έλεγχος εγκυρότητας).

Δ2. Για κάθε μήνα, να εμφανίζει το συνολικό αριθμό ατόμων που ενημέρωσαν οι δέκα (10) εθελοντές. Ο υπολογισμός του συνολικού αριθμού ατόμων, που ενημέρωσαν κάθε μήνα, να γίνει με κλήση κατάλληλης συνάρτησης.

Δ3. Να εμφανίζει τα ονόματα των τριών εθελοντών που ενημέρωσαν τα περισσότερα άτομα, κατά τη διάρκεια του προηγούμενου έτους.

Να θεωρήσετε ότι κάθε εθελοντής ενημέρωσε διαφορετικό συνολικό αριθμό ατόμων κατά τη διάρκεια του έτους.

Δ4. Να κατασκευάσετε τη συνάρτηση του ερωτήματος Δ2.

Να θεωρήσετε ότι κάθε άτομο ενημερώνεται μόνο από ένα εθελοντή.

ΑΣΚΗΣΗ 1

```
XOR=1000
min1=1000
ON1=""
ON2=""
min2=1000
S1=0
S=0

M=input("Δώσε μέγεθος αρχείου")

while M<XOR:
   ON=raw_input("Δώσε ονομα αρχείου")
   print "Επιτρεπτή αποθήκευση"
if M>10:
      S1=S1+1
   S=S+1

   XOR=XOR-M

   if M<min1:
      min2=min1
      ON2=ON1
      min1=M
      ON1=ON

   elif M<min2:
      min2=M
ON2=ON

M=input("Δώσε μέγεθος αρχείου")

print "ποσοστό με μέγεθος μεγαλύτερο των 10GB", S1*100/S

print "Τα δύο μικρότερα αρχεία είναι τα", ON1, "και", ON2
```

ΑΣΚΗΣΗ 2

```
def SUM(A,m):
   S=0
   for x in range(10):
      S=S+A[x] [y]

   return S

ON=[]   ;  A=[]

for x in range(10):
   on=str(raw_input("Δώσε όνομα"))
   ON.append(on)
   T=[]
   for y in range(12):
print "Μηνας:",y+1
      a=input("Δώσε αριθμό ατόμων")
T.append(a)
   A.append(T)

for y in range(12):
print "Για τον μήνα:", y+1, "σύνολο ατόμων ", SUM(A,y)

max1=0
max2=0
max3=0
on1=””
on2=””
on3=””
for x in range(10):
   S=0
   for y in range(12):
      S=S+A[x] [y]

   if S>max1:
      max3=max2
      max2=max1
      max1=S
      on3=on2
      on2=on1
      on1=ON[x]

elif max2==0 or S>max2:
      max3=max2
      max2=S
      on3=on2
      on2=ON[x]

   if max3==0 or S>max3:
max3=S
      on3=ON[x]

print "τρείς εθελοντές που ενημέρωσαν τα περισσότερα άτομα", on1,on2,on3
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2016 - ΠΑΛΑΙΟ ΣΥΣΤΗΜΑ

1) Ένα ξενοδοχείο χρεώνει την ενοικίαση των δωματίων του ανάλογα με τον αριθμό των ημερών ενοικίασης και την τουριστική περίοδο, σύμφωνα με τον παρακάτω πίνακα:

ΑΡΙΘΜΟΣ ΗΜΕΡΩΝ	**ΤΟΥΡΙΣΤΙΚΗ ΠΕΡΙΟΔΟΣ**	
	ΧΑΜΗΛΗ	**ΥΨΗΛΗ**
1-3	40€ ανά ημέρα	70€ ανά ημέρα
4-7	30€ ανά ημέρα	55€ ανά ημέρα
>7	25€ ανά ημέρα	50€ ανά ημέρα

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Γ 1. Για καθεμιά από τις 500 κρατήσεις του ξενοδοχείου κατά τοπροηγούμενο έτος:

α. Να διαβάζει τον αριθμό των ημερών ενοικίασης καθώς και την τουριστική περίοδο που έγινε η κράτηση, εξασφαλίζοντας ότι η επιτρεπτή τιμή για την τουριστική περίοδο είναι ΧΑΜΗΛΗ ή ΥΨΗΛΗ.

β. Να καλεί υποπρόγραμμα με είσοδο τον αριθμό των ημερών ενοικίασης και την τουριστική περίοδο, το οποίο να υπολογίζει, με βάση τον προηγούμενο πίνακα, τη χρέωση της κράτησης. Ο υπολογισμός της χρέωσης δεν γίνεται κλιμακωτά.

γ. Να εμφανίζει τη χρέωση της κράτησης.

Γ2. Να υπολογίζει και να εμφανίζει τη συνολική χρέωση των κρατήσεων τουξενοδοχείου για καθεμιά τουριστική περίοδο του προηγούμενου έτους.

Γ3. Να κατασκευάσετε το υποπρόγραμμα του ερωτήματος Γ1.β.

2) Μια εταιρεία έχει δύο υποκαταστήματα, ένα στην Αθήνα και ένα στη Θεσσαλονίκη. Σε κάθε υποκατάστημα εργάζονται 10 πωλητές.

Να αναπτύξετε αλγόριθμο σε Python, ο οποίος:

Δ1. Για καθέναν από τους 20 πωλητές της εταιρείας, να διαβάζει το όνομά του και να το καταχωρίζει την λίστα ΟΝ. Να διαβάζει την πόλη του υποκαταστήματος και να δημιουργεί τον κωδικό τον οποί καταχωρεί στην λίστα C.

Ο κωδικός δημιουργείται αυτόματα έτσι ώστε ο κωδικός των πωλητών του υποκαταστήματος της Αθήνας ξεκινάει με το γράμμα Α και ακολουθεί ακέραιος αριθμός και ο κωδικός των πωλητών της Θεσσαλονίκης ξεκινάει με το γράμμα Β και ακολουθεί ακέραιος αριθμός. Να θεωρήσετε ότι όλα τα ονόματα και όλοι οι κωδικοί είναι διαφορετικοί μεταξύ τους.

Δ2. Για κάθε παραγγελία της εταιρείας στη διάρκεια του προηγούμενου έτους, να διαβάζει τον κωδικό του πωλητή. Αν ο κωδικός ανήκει σε πωλητή της εταιρείας, να διαβάζει το ποσό της αντίστοιχης παραγγελίας που πήρε ο πωλητής (δεν απαιτείται έλεγχος εγκυρότητας) τον οποί και καταχωρεί στην λίστα Ρ (σύνολο παραγγελιών κάθε πωλητή) ή, διαφορετικά, να εμφανίζει το μήνυμα «Άγνωστος κωδικός». Η επαναληπτική διαδικασία να τερματίζεται όταν δοθεί, ως κωδικός πωλητή, η τιμή ΤΕΛΟΣ.

Δ3. Να υπολογίζει τις συνολικές πωλήσεις κάθε πωλητή στη διάρκεια του προηγούμενου έτους και να τις εμφανίζει μαζί με το όνομά του. Να θεωρήσετε ότι κάθε πωλητής πήρε παραπάνω από μία παραγγελία στη διάρκεια του προηγούμενου έτους.

Δ4. Για κάθε υποκατάστημα να βρίσκει και να εμφανίζει τα ονόματα των τριών πωλητών με τις μεγαλύτερες συνολικές πωλήσεις στη διάρκεια του προηγούμενου έτους. Να θεωρήσετε ότι οι συνολικές πωλήσεις όλων των πωλητών είναι διαφορετικές μεταξύ τους.

ΑΣΚΗΣΗ 1

```
def ypologismod(H,P):
   if P=="ΧΑΜΗΛΗ":
      if H>=1 and H<=3:
         K=H*40
      if H>=3 and H<=7:
         K=H*30
      if H>7:
         K=H*25
   if P=="ΥΨΗΛΗ":
      if H>=1 and H<=3:
         K=H*70
      if H>=3 and H<=7:
         K=H*55
      if H>7:
         K=H*50

   return K

S1=0
S2=0
for x in range (500):
   H=input("Δώσε αριθμό ημερών ενοικίασης")
P=raw_input("Δώσε την τουριστική περίοδο ")
while P not in ["ΧΑΜΗΛΗ", "ΥΨΗΛΗ"]:
      P=raw_input("Δώσε ΧΑΜΗΛΗ ή ΥΨΗΛΗ")

print "χρέωση της κράτησης",ypologismod(H,P)

if P=="ΧΑΜΗΛΗ":
      S1=S1+ypologismod(H,P)

   if P=="ΥΨΗΛΗ":
      S2=S2+ypologismod(H,P)

print "συνολική χρέωση ΧΑΜΗΛΗΣ περιόδου", S1

print "συνολική χρέωση ΥΨΗΛΗΣ περιόδου", S2
```

ΑΣΚΗΣΗ 2

```
P=[]
ON=[]
C=[]

for x in range(20):
   P.append(0)

   on=str(raw_input("Δώσε ονομα πωλητή"))

   POLH=raw_input("Δώσε πόλη")
   if POLH=="ΑΘΗΝΑ":
      c.append("A"+str(x))
   if POLH=="ΘΕΣΣΑΛΟΝΙΚΗ":
      c.append("B"+str(x))

code=raw+input("Δώσε κωδικό")
while code not in C:
   print "Άγνωστος κωδικός"
   code=raw+input("Δώσε κωδικό")

while code!="ΤΕΛΟΣ":

   p=input("Δώσε παραγγελία")
   for x in range(20):
      if C[x]==code:
         P[x]=P[x]+p

   code=raw+input("Δώσε κωδικό")
   while code not in C:
      print "Άγνωστος κωδικός"
      code=raw+input("Δώσε κωδικό")

for x in range(20):
   print ON[x], P[x]

N=20
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if P[j]>P[j-1]:
         P[j],P[j-1]=P[j-1],P[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         C[j],C[j-1]=C[j-1],C[j]

print ON[0], ON[1], ON[2]
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2016 – ΟΜΟΓΕΝΕΙΣ ΕΞΩΤΕΡΙΚΟΥ

1) Στον Πανελλήνιο Διαγωνισμό Πληροφορικής συμμετέχουν Έλληνες μαθητές από τις πέντε ηπείρους. Να κατασκευάσετε πρόγραμμα σε Python, το οποίο:

Γ2. Να εισάγει σε λίστα Η πέντε (5) θέσεων τα ονόματα των ηπείρων, με την εξής σειρά: Ευρώπη, Αφρική, Ασία, Ωκεανία, Αμερική.

Γ3. Για κάθε μαθητή που συμμετέχει στο διαγωνισμό, να διαβάζει την ήπειρο από την οποία προέρχεται, με τη μορφή αριθμού, ως εξής: 1 για την Ευρώπη, 2 για την Αφρική, 3 για την Ασία, 4 για την Ωκεανία και 5 για την Αμερική. Η εισαγωγή να τερματίζεται όταν δοθεί ο αριθμός 0.

(Δεν απαιτείται έλεγχος εγκυρότητας.)

Γ4. Να υπολογίζει τον αριθμό των μαθητών που συμμετέχουν από κάθε ήπειρο.

Γ5. Να εμφανίζει τα ονόματα των πέντε (5) ηπείρων και δίπλα από κάθε όνομα, τον αριθμό των μαθητών που συμμετέχουν από αυτή την ήπειρο. Τα στοιχεία να είναι ταξινομημένα σε φθίνουσα σειρά με βάση τον αριθμό των μαθητών.

Σημείωση: Να θεωρήσετε ότι οι αριθμοί των μαθητών, που συμμετέχουν από τις πέντε ηπείρους, είναι όλοι διαφορετικοί μεταξύ τους.

2) Στην Ελλάδα υπάρχουν 41 Κέντρα Περιβαλλοντικής Εκπαίδευσης (ΚΠΕ), τα οποίαδέχονται οργανωμένες επισκέψεις μαθητών. Να κατασκευάσετε πρόγραμμα σε Python το οποίο:

Δ2. Για κάθε ΚΠΕ να διαβάζει:

α. το όνομά του και να το καταχωρίζει σε κατάλληλη λίστα.

β. τον αριθμό των επισκέψεων, που δέχτηκε για κάθε μήνα ενός έτους, και να τον καταχωρίζει σε κατάλληλη λίστα.

(Δεν απαιτείται έλεγχος εγκυρότητας.)

Δ3. Να εμφανίζει το όνομα του ΚΠΕ με το μεγαλύτερο συνολικό ετήσιο αριθμό επισκέψεων. Να θεωρήσετε ότι ένα μόνο ΚΠΕ έχει το μεγαλύτερο συνολικόετήσιο αριθμό επισκέψεων.

Δ4. Να εμφανίζει τον συνολικό αριθμό επισκέψεων, που δέχτηκαν όλα τα ΚΠΕ την άνοιξη (δηλ. κατά τους μήνες 3, 4 και 5).

Δ5. Να εμφανίζει τους αριθμούς των μηνών του έτους, κατά τους οποίους και τα 41 ΚΠΕ δέχτηκαν επισκέψεις.

ΑΣΚΗΣΗ 1

```
H=[]
A=[]
for x in range (5):
   on=str(raw_input("Δώσε ονομα Ηπείρου"))

   H.append(on)
   A.append(0)

c1=0
h=input("Δώσε Ηπειρο")
while H!=0:

   A[c1]=A[c1]+1

   h=input("Δώσε Ηπειρο")
   c1=c1+1

N=5

for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] > A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         H[j],H[j-1]=H[j-1],H[j]

for x in range(5):
   print H[x], A[x]
```

ΑΣΚΗΣΗ 2

```
ON=[]

E=[]

for x in range (41):
   on=str(raw_input("Δώσε ονομα ΚΠΕ"))
   ON.append(on)
   T=[]
   for y in range(12):
      print "ΜΗΝΑΣ:",y+1
      e=input("Δώσε αριθμό επισκέψεων")
      T.append(e)
   E.append(T)

max1=0
for x in range (41):
   S=0
   SA=0
   for y in range(12):
      S=S+E[x][y]

      if y==3 or y==4 or y==5:
         SA=SA+E[x][y]
   if S>max1:
      max1=S
      onoma=ON[x]

print "ΚΠΕ με το μεγαλύτερο αριθμό επισκέψεων",onoma

print "αριθμό επισκέψεων που δέχτηκαν την άνοιξη ",SA

for y in range(12):
   F=False
   for x in range (41):
      if E[x][y]==0:
         F=True
   if F==False:
      print "Ο Μήνας:", y+1
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2017

1) Στο πλαίσιο ενός τοπικού σχολικού πρωταθλήματος βόλεϊ συμμετέχουν 5 σχολεία, αριθμημένα από το 1 έως το 5. Κάθε σχολείο παίζει μία φορά με όλα τα υπόλοιπα. Άρα θα πραγματοποιηθούν συνολικά 10 αγώνες. Νικητής ενός αγώνα είναι το σχολείο που έχει κερδίσει 3 σετ. Ο νικητής παίρνει 2 βαθμούς και ο ηττημένος 1 βαθμό.

Κάθε αγώνας προσδιορίζεται από τα σχολεία που παίζουν μεταξύ τους και το αποτέλεσμα του αγώνα σε σετ. Για παράδειγμα , η σειρά των στοιχείων : 4, 5, 1 , 3σημαίνει ότι το σχολείο 4 έπαιξε με το σχολείο 5 και έχασε τον αγώνα με 1 σετ υπέρ και 3 κατά . Αυτό αντίστοιχα σημαίνει ότι το σχολείο 5 κέρδισε τον αγώνα με το σχολείο 4 με 3 σετ υπέρ και 1 σετ κατά.

Τα δεδομένα των αγώνων αποθηκεύονται σε τρείς λίστες : Στην Α καταχωρούνται η βαθμολογία κάθε σχολείου (το άθροισμα των βαθμών του), στην λίστα Βτο άθροισμα των σετ υπέρ και στην λίστα Στο άθροισμα των σετ κατά , από όλους τους αγώνες.

Να κατασκευάσετε πρόγραμμα σε Python το οποίο :

Γ1. Να διαβάζει τα ονόματα των 5 σχολείων και να τα καταχωρίζει στην λίστα ΟΝ. Η σειρά των σχολείων καθορίζει την αρίθμησή τους (1 έως 5).

Γ2. Να διαβάζει για κάθε αγώνα τη σειρά των 4 στοιχείων που τον προσδιορίζουν και να ενημερώνει τις λίστες A, B, C και για τα δύο σχολεία όπως περιγράφεται παραπάνω.

Γ3. Να κατατάσσει τα σχολεία σε φθίνουσα σειρά ανάλογα με τη βαθμολογία τους και σε περίπτωση ισοβαθμίας να προηγείται το σχολείο με τα περισσότερα σετ υπέρ.

Γ4. Να εμφανίζει τα ονόματα των σχολείων, τη βαθμολογία τους, το άθροισμα των σετ υπέρ και το άθροισμα των σετ κατά, με βάση τη σειρά κατάταξής τους.

Σημείωση : Θεωρείστε ότι δεν υπάρχει περίπτωση δύο σχολεία να έχουν και την ίδια βαθμολογία και τον ίδιο αριθμό σετ υπέρ.

2) Σε ένα σεμινάριο διάρκειας 6 μηνών, τηρούνται απουσίες ανά μήνα για κάθε συμμετέχοντα. Στο σεμινάριο συμμετέχουν 50 επιμορφούμενοι και ο καθένας έχει ένα μοναδικό αλφαριθμητικό κωδικό, που αποθηκεύεται στην λίστα Κ. Οι απουσίες κάθε συμμετέχοντα ανά μήνα σεμιναρίου αποθηκεύονται σε λίστες απουσιών Α1, Α2, Α3, Α4, Α5, Α6. Η γραμματεία τηρεί το σύνολο των απουσιών για τα δύο τρίμηνα του εξαμήνου σε λίστες ΑΤ1 και ΑΤ2, όπου η πρώτη προσδιορίζει το πρώτο τρίμηνο και η δεύτερη το δεύτερο τρίμηνο για κάθε συμμετέχοντα.

Να κατασκευάσετε πρόγραμμα σε Python αποτελούμενο από υποπρογράμματα ως εξής :

Δ1. Διαδικασία EIS, που διαβάζει τον κωδικό του κάθε επιμορφούμενου, τις απουσίες του ανά μήνα σεμιναρίου και ενημερώνει την λίστα Κ και τις λίστες απουσιών (θεωρείστε ότι τα δεδομένα εισάγονται σωστά).

Δ2. Συνάρτηση ΑΝΑΖ, που δέχεται τον κωδικό ενός επιμορφούμενου και την λίστα Κ των κωδικών και επιστρέφει τον αριθμό της θέσης που βρίσκεται ο κωδικός που αναζητείται. Αν ο κωδικός δεν βρεθεί, επιστρέφει 0.

Δ3. Συνάρτηση ΣΥΝΑΡ, που υπολογίζει το σύνολο απουσιών για έναν επιμορφούμενο σε ένα τρίμηνο. Η συνάρτηση δέχεται τις λίστες απουσιών και επιστρέφει ενημερωμένες τις λίστες των απουσιών του κάθε τριμήνου.

Δ4. Κύριο πρόγραμμα το οποίο:

α) καλεί τη διαδικασία EIS για είσοδο δεδομένων.

β) για κάθε επιμορφούμενο υπολογίζει το σύνολο των απουσιών των δύοτριμήνων καλώντας τη συνάρτηση ΣΥΝΑΡ και ενημερώνει τις λίστες ΑΤ1 και ΑΤ2.

γ) διαβάζει επαναληπτικά έναν κωδικό. Για τον συγκεκριμένο κωδικό καλείται η συνάρτηση ΑΝΑΖ. Αν ο κωδικός αντιστοιχεί σε επιμορφούμενο, να εμφανίζει κατάλληλο μήνυμα δυνατότητας ή μη συμμετοχής του στις εξετάσεις. Στις εξετάσεις δικαιούνται συμμετοχής οι επιμορφούμενοι που έχουν λιγότερες από 10 απουσίες σε καθένα από τα δύο τρίμηνα. Αν ο κωδικός δεν βρεθεί, εμφανίζει μήνυμα «ΔΕΝ ΒΡΕΘΗΚΕ Ο ΚΩΔΙΚΟΣ». Η διαδικασία επαναλαμβάνεται μέχρι να δοθεί ως κωδικός η λέξη ΤΕΛΟΣ.

ΑΣΚΗΣΗ 1

```
ON=[]  ;  A=[]  ;  B=[]  ; C=[]

for x in range(5):
   on=str(raw_input("Δώσε ονομα"))
   ON.append(on)

   A.append(0)
   B.append(0)
   C.append(0)

for x in range(10):
   on1=raw_input("Δώσε ονομα σχολείου 1")
on2=raw_input("Δώσε ονομα σχολείου 2")

setY=input("Δώσε σετ υπέρ")
setK=input("Δώσε σετ κατά")

for x in range(5):
      if on1==ON[x]:
         Ton1=x
      if on2==ON[x]:
         Ton2=x

   B[Ton1]=B[Ton1]+setY
   C[Ton1]=C[Ton1]+setK
   B[Ton2]=B[Ton2]+setK
   C[Ton2]=C[Ton2]+setY

   if setY==3:
      A[Ton1]=A[Ton1]+2
      A[Ton2]=A[Ton2]+1
   elif setK==3:
      A[Ton1]=A[Ton1]+1
      A[Ton2]=A[Ton2]+2

N=len(A)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] > A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]
         C[j],C[j-1]=C[j-1],C[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]

      if  A[j]==A[j-1]  and  B[j]>B[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]
         C[j],C[j-1]=C[j-1],C[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]

for x in range(5):
   print ON[x], A[x], B[x], C[x]
```

ΑΣΚΗΣΗ 2

```
def EIS(K,A1,A2,A3,A4,A5,A6):
   k=str(raw_input("Δώσε κωδικό")
   K.append(k)
   a1=input("Δώσε απουσίες μήνα 1")
   a2=input("Δώσε απουσίες μήνα 2")
   a3=input("Δώσε απουσίες μήνα 3")
   a4=input("Δώσε απουσίες μήνα 4")
   a5=input("Δώσε απουσίες μήνα 5")
   a6=input("Δώσε απουσίες μήνα 6")

   A1.append(a1)
   A2.append(a2)
   A3.append(a3)
   A4.append(a4)
   A5.append(a5)
   A6.append(a6)

   return K,A1,A2,A3,A4,A5,A6

def ANAZ(Kod, K):
   F=False
   for x in range(K):
        if K[x]==Kod:
           returt x
           F=True
   if F==False:
        return 0

def SYNAP(A1,A2,A3,A4,A5,A6, AT1,AT2):
   for x in range(50):
      AT1.append(A1[x]+A2[x]+A3[x])
      AT2.append(A4[x]+A5[x]+A6[x])
   return AT1,AT2

K=[]  ;  AT1=[] ;  AT2=[]
A1=[] ;  A2=[] ; A3=[] ;  A4=[] ;  A5=[] ; A6=[]

for x in range(50):
  K, A1,A2,A3,A4,A5,A6 = EIS(K, A1,A2,A3,A4,A5,A6)

AT1,AT2=SYNAP(A1,A2,A3,A4,A5,A6, AT1,AT2)

Kod=raw_input("Δώσε κωδικό")
while kod!="ΤΕΛΟΣ":
   Z=ANAZ(Kod, K)
   if Z!=0:
      if AT1[Z] <10 and AT2[Z]<10:
         print "δικαιούται συμμετοχή"
      else:
         print "ΔΕΝ δικαιούται συμμετοχή
   else:
       print "ΔΕΝ ΒΡΕΘΗΚΕ Ο ΚΩΔΙΚΟΣ"
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2017– ΟΜΟΓΕΝΕΙΣ ΕΞΩΤΕΡΙΚΟΥ

1) Σε μια έκθεση αποδήμου ελληνισμού χρησιμοποιείται αίθουσα χωρητικότητας 1000 ατόμων. Στην αίθουσα εγκαταστάθηκε ηλεκτρονικό σύστημα διαχείρισης εισόδου-εξόδου επισκεπτών, το οποίο λειτουργεί ως εξής:

Κάθε φορά που γίνεται είσοδος επισκεπτών εισάγεται η τιμή 1, ενώ κάθε φορά που γίνεται έξοδος επισκεπτών εισάγεται η τιμή 2. Για τον τερματισμό της λειτουργίας του συστήματος εισάγεται η τιμή 0.

Η είσοδος πραγματοποιείται είτε μεμονωμένα είτε σε ομάδες. Προκειμένου να επιτραπεί η είσοδος, ζητείται ο αριθμός επισκεπτών που θέλουν να εισέλθουν και, εφόσον η ενδεχόμενη είσοδός τους δεν υπερβαίνει το όριο χωρητικότητας της αίθουσας, τότε επιτρέπεται· διαφορετικά, απορρίπτεται με κατάλληλο μήνυμα.

Η έξοδος πραγματοποιείται μεμονωμένα, δηλαδή ένα άτομο κάθε φορά. Ο τερματισμός επιτρέπεται, όταν η αίθουσα είναι άδεια.

Για την υποστήριξη του συστήματος να αναπτύξετε πρόγραμμα το οποίο:

Γ1. Να διαβάζει τον κωδικό επιθυμητής λειτουργίας (1 για είσοδο, 2 για έξοδο και 0 για τερματισμό), μέχρι τον τερματισμό της λειτουργίας του συστήματος.

Γ2. α. Στην περίπτωση που δοθεί ο κωδικός 1, να διαβάζει τον αριθμό των ατόμων και με τη χρήση της λογικής συνάρτησης ΙΝ να ελέγχει αν επιτρέπεται η είσοδός τους. Αν η είσοδός τους επιτρέπεται, εισέρχονται στην αίθουσα· διαφορετικά, εμφανίζεται το μήνυμα ΔΟΚΙΜΑΣΤΕ ΑΡΓΟΤΕΡΑ.

β. Στην περίπτωση που δοθεί ο κωδικός 2, θεωρείται ότι εξέρχεται ένα άτομο. Η εκτέλεση της συγκεκριμένης λειτουργίας να επιτρέπεται, όταν η αίθουσα δεν είναι κενή· διαφορετικά, να εμφανίζει το μήνυμα ΑΔΥΝΑΤΗ ΛΕΙΤΟΥΡΓΙΑ.

Γ3. Μετά τον τερματισμό να εμφανίζει τον συνολικό αριθμό των επισκεπτών.

Γ4. Να αναπτύξετε τη λογική συνάρτηση ΙΝ.

(Να θεωρήσετε ότι δεν απαιτείται έλεγχος εγκυρότητας για τις τιμές εισόδου και ότι η αίθουσα είναι αρχικά κενή).

2) Στο τελευταίο φεστιβάλ ψηφιακής δημιουργίας συμμετείχαν 10 ομάδες μαθητών. Κάθε ομάδα παρουσίασε μια εργασία. Από κάθε ομάδα ζητήθηκε να βαθμολογήσει όλες τις εργασίες, τόσο τη δική της όσο και των υπολοίπων 9 ομάδων.

Να κατασκευάσετε πρόγραμμα σε Python το οποίο:

Δ1. Να καταχωρίζει:

α. τα ονόματα των ομάδων, σε λίστα ΟΝ.

β. τους ακέραιους βαθμούς, σε λίστα Β. Κάθε στοιχείο της λίστας Β είναι οι 10 βαμολογίες κάθε ομάδας. Οι βαθμοί να εισάγονται, για κάθε ομάδα με τη σειρά, από την πρώτη μέχρι τη δέκατη.

Δ1. Να εμφανίζει το όνομα της ομάδας που συγκέντρωσε τον μεγαλύτερο μέσο όρο βαθμολογίας.

Δ3. Να εμφανίζει το όνομα της ομάδας η οποία βαθμολόγησε τον εαυτό της πλησιέστερα στον μέσο όρο των βαθμών που έλαβε.

(Για τα ερωτήματα Δ3 και Δ4 να θεωρήσετε ότι η τιμή του μέσου όρου είναι μοναδική).

ΑΣΚΗΣΗ 1

```
def IN(A,E):
   if A>E:
      print "ΔΟΚΙΜΑΣΤΕ ΑΡΓΟΤΕΡΑ"
   else:
      return True

E=1000
S=0

K=input("Δώσε κωδικό επιθυμητής λειτουργίας")

while K!=0:

   if K==1:
      A=input("Δώσε αριθμό ατόμων")
      if IN(A,E)==True:
         E=E-A
         S=S+A
   if K==2 :
      if E!=1000:
         E=E+1
      else:
         print "ΑΔΥΝΑΤΗ ΛΕΙΤΟΥΡΓΙΑ"

   K=input("Δώσε κωδικό επιθυμητής λειτουργίας")
```

ΑΣΚΗΣΗ 2

```
ON=[]
B=[]
D=[]

for x in range(10):
   on=str(raw_input("Δώσε όνομα"))
   ON.append(on)

for x in range(10):
   T=[]
   for y in range(10):
      print ON[x]
      V=input("Δώσε βαθμό")
      T.append(V)
   B.append(T)

max1=0
for x in range(10):
   S=0.0
   for y in range(10):
      S=S+B[x][y]

   MO=S/10
   if MO>max1:
      max1=MO
      onoma=ON[x]

print "ομάδας με μεγαλύτερο μέσο όρο",onoma


for x in range(10):
   S=0
   for y in range(10):
      S=S+B[y][x]

   MO=S/10
   D.append(MO-B[x][x])

min1=D[0]
ON1=ON[0]

for x in range(1,10):
   if D[x]<min1:
      min1=D[x]
      ON1=ON[x]

print "ομάδας που βαθμολόγησε τον εαυτό της πλησιέστερα στον μέσο όρο",ON1
```

ΥΠΟΛΕΙΠΟΜΕΝΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2017

1) Σε μια εξέταση του μαθήματος της Αγγλικής Γλώσσας εξετάζονται πενήντα (50) μαθητές προφορικά και γραπτά. Οι μαθητές βαθμολογούνται από το 0.0 έως και το 20.0 σε κάθε εξέταση (προφορικά, γραπτά).

Να γράψετε ένα πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω:

Γ1. Σε δομή επανάληψης να διαβάζει το ονοματεπώνυμο, την προφορική και τη γραπτή βαθμολογία κάθε μαθητή. Δεν απαιτείται έλεγχος ορθότητας εισαγωγής τιμών.

Γ2. Να εμφανίζει τα ονοματεπώνυμα των μαθητών που έχουν άθροισμα προφορικής και γραπτής βαθμολογίας μεγαλύτερο από το 19.5.

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος των μαθητών που η γραπτή βαθμολογία τους είναι μεγαλύτερη από την προφορική τους.

Γ4. Να υπολογίζει και να εμφανίζει τον μέσο όρο της γραπτής βαθμολογίας και τον μέσο όρο της προφορικής βαθμολογίας όλων των μαθητών.

2) Το Υπουργείο Πολιτισμού διατηρεί στατιστικά στοιχεία για το θέατρο της Αρχαίας Επιδαύρου σχετικά με τον τίτλο κάθε παράστασης και το πλήθος των θεατών που την παρακολούθησαν (κάθε παράσταση παρουσιάζεται μόνο μία φορά και υπάρχει τουλάχιστον μία παράσταση).

Να γράψετε ένα πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω:

Δ1. Να διαβάζει τον τίτλο κάθε παράστασης και το πλήθος των θεατών που την παρακολούθησαν. Τα στοιχεία αυτά να καταχωρίζονται στις λίστες με ονόματα PAR και S_P αντίστοιχα. Να γίνεται έλεγχος ορθότητας για το πλήθος των θεατών που εισάγεται έτσι ώστε να είναι θετικός αριθμός. Η εισαγωγή των στοιχείων θα τερματίζεται όταν δοθεί ως τίτλος παράστασης η λέξη «TELOS».

Δ2. Να εντοπίζει και να εμφανίζει τον τίτλο της παράστασης με το μέγιστο πλήθος θεατών. Να θεωρήσετε ότι υπάρχει μία μόνο τέτοια παράσταση.

Δ3. Να υπολογίζει και εμφανίζει τον μέσο όρο των θεατών όλων των παραστάσεων.

Δ4. Κάθε παράσταση με πλήθος θεατών μεγαλύτερο ή ίσο από 1000 άτομα επιδοτείται με 10000€, ενώ κάθε παράσταση με πλήθος θεατών μικρότερο των 1000 ατόμων επιδοτείται με 5000€. Να υπολογίσετε και να εμφανίσετε το συνολικό ποσό της επιδότησης που θα διαθέσει το Υπουργείο Πολιτισμού.

ΑΣΚΗΣΗ 1

```
c=0
SG=0
SP=0
for x in range(50):
   on=raw_input("Δώσε όνομα")

   G=float(input("Δώσε βαθμό γραπτά"))

   P=float(input("Δώσε βαθμό προφορικά"))

   if G+P>19.5 :
      print on

   if G>P:
      c=c+1

   SG=SG+G
   SP=SP+P

print "έχουν γραπτή βαθμολογία μεγαλύτερη από την προφορική",c

print "Μέσος όρος γραπτών",SG/50
print "Μέσος όρος προφορικών",SP/50
```

ΑΣΚΗΣΗ 2

```
PAR=[]  ;  S_P=[]

T=str(raw_input("Δώσε τίτλο παράστασης "))
while T!="TELOS":
   P=input("Δώσε πλήθος θεατών ")
   PAR.append(T)
   S_P.append(P)
   T=str(raw_input("Δώσε τίτλο παράστασης "))

max1=S_P[0]
on=PAR[0]
for x in range(1,len(PAR)):
   if S_P[x]>max1:
      max1=S_P[x]
      on=PAR[x]
print "παράστασης με το μέγιστο πλήθος θεατών",on

c1=0
S=0.0
for x in S_P:
   S=S+x
   if x>=1000:
      c1+=1

print "μέσο όρο των θεατών όλων των παραστάσεων",S/len(S_P)
print "συνολικό ποσό της επιδότησης ", c1*10000 + (len(S_P)-c1)*5000
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2018

1) Ένα λιμάνι διαθέτει αποθηκευτικό χώρο χωρητικότητας 170 εμπορευματοκιβωτίων (containers). Σε καθημερινή βάση, στο τέλος της ημέρας, καταχωρίζεται ο αριθμός των εμπορευματοκιβωτίων που έχουν εισέλθει και εξέλθει από αυτόν.

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Δ1. Να διαβάζει για κάθε ημέρα το συνολικό πλήθος εμπορευματοκιβωτίων που εισήλθαν, καθώς και το συνολικό πλήθος εκείνων που εξήλθαν από τον αποθηκευτικό χώρο. Οι τιμές που διαβάζονται να ελέγχονται ώστε ο αριθμός των εμπορευματοκιβωτίων που παραμένουν στον αποθηκευτικό χώρο στο τέλος της ημέρας να είναι από 0 μέχρι και 170. Σε αντίθετη περίπτωση να θεωρούνται λανθασμένες και να επανεισάγονται.

Δ2. Για τον τερματισμό της εισαγωγής δεδομένων το πρόγραμμα εμφανίζει το μήνυμα "Τέλος Εισαγωγής Στοιχείων; ΝΑΙ / ΟΧΙ". Αν εισαχθεί η τιμή "ΝΑΙ", να τερματίζεται η εισαγωγή δεδομένων.

Δ3. Να βρίσκει και να εμφανίζει τον μέγιστο ημερήσιο αριθμό εισερχόμενων εμπορευματοκιβωτίων.

Δ4. Να υπολογίζει και να εμφανίζει τη μέση ημερήσια διακίνηση εμπορευματοκιβωτίων. Η ημερήσια διακίνηση είναι το άθροισμα του πλήθους των εισερχομένων και των εξερχομένων εμπορευματοκιβωτίων της ημέρας.

Δ5. Να υπολογίζει και να εμφανίζει το πλήθος των ημερών που παρέμειναν στον αποθηκευτικό χώρο τουλάχιστον 10 εμπορευματοκιβώτια, στο τέλος κάθε ημέρας.

Δ6. Να υπολογίζει και να εμφανίζει τον μέσο όρο του πλήθους των εμπορευματοκιβωτίων που παρέμειναν στον αποθηκευτικό χώρο, στο τέλος κάθε ημέρας, από την έναρξη μέχρι τον τερματισμό εισαγωγής δεδομένων.

Σημειώσεις Να θεωρήσετε ότι :

α) Αρχικά ο αποθηκευτικός χώρος είναι κενός.

β) Οι αριθμοί που εισάγονται για το πλήθος των εισερχομένων και των εξερχομένων εμπορευματοκιβωτίων είναι μεγαλύτεροι ή ίσοι του 0.

γ) Υπάρχει καταχώριση στοιχείων για τουλάχιστον μια ημέρα.

2) Ο φορέας διαχείρισης μιας περιοχής οικολογικού ενδιαφέροντος, προκειμένου να εκτιμήσει την ποιότητα των υδάτων των ποταμών της περιοχής, πραγματοποιεί μία δειγματοληψία τον μήνα σε κάθε

ποταμό στη διάρκεια ενός έτους. Το δείγμα νερού αναλύεται και ανιχνεύονται οι ρύποι . Η επικινδυνότητα ενός ρύπου εκφράζεται με έναν ακέραιο αριθμό από το 1 έως και το 10. Στην κλίμακα αυτή η μεγαλύτερη τιμή αντιστοιχεί σε υψηλότερη επικινδυνότητα. Ένας δείκτης της επικινδυνότητας των υδάτων είναι η επικινδυνότητα εκείνου του ρύπου που έχει τη μέγιστη τιμή .

Να αναπτύξετε κύριο πρόγραμμα σε Python το οποίο:

Δ1. α. Να διαβάζει το πλήθος των ποταμών της περιοχής, ελέγχοντας ότι δεν δίνεται τιμή μεγαλύτερη του 20 .

β. Να διαβάζει τα ονόματα των ποταμών αυτών και να τα καταχωρίζει σε διαδοχικές θέσεις της λίστας ΟΝ.

Δ2. Για κάθε δειγματοληψία: να εμφανίζει το όνομα καθενός ποταμού της περιοχής και να υπολογίζει την επικινδυνότητά του καλώντας το υποπρόγραμμα Υ_Ε (που θα κατασκευάσετε στο ερώτημα Δ4) . Την επικινδυνότητα αυτή να την καταχωρίζει κατάλληλα σε λίστα ΕΡ.

Δ3. Να εμφανίζει τα ονόματα των ποταμών στους οποίους ο μέσος όρος επικινδυνότητας στη διάρκεια του έτους, κυμάνθηκε πάνω από 7. Αν δεν υπάρχει κανένας ποταμός που να ικανοποιεί το κριτήριο αυτό, να εμφανίζεται κατάλληλο μήνυμα.

Να αναπτύξετε το υποπρόγραμμα Υ_ Ε το οποίο:

Δ4. α) Να διαβάζει διαδοχικά τις τιμές της επικινδυνότητας κάθε ρύπου που βρέθηκε. Η εισαγωγή να τερματίζεται όταν δοθεί η τιμή 0 (που σημαίνει ότι δεν υπάρχει άλλος ρύπος) .

β) Να επιστρέφει τη μέγιστη τιμή επικινδυνότητας από τις τιμές που διάβασε.

Σημείωση

α) Δεν απαιτούνται επιπλέον έλεγχοι εγκυρότητας τιμών εκτός από αυτόν που ζητείται στο ερώτημα Δ1.α .

β) Να θεωρήσετε ότι υπάρχει τουλάχιστον ένας ποταμός.

γ) Να θεωρήσετε ότι σε κάθε δειγματοληψία υπάρχει τουλάχιστον ένας ρύπος.

ΑΣΚΗΣΗ 1

```
ES=[]
EK=[]
A=[]
AP=0
D="ΟΧΙ"

while D!="ΝΑΙ":

   es=input('Δωσε εισερχόμενα')
   ek=input('Δωσε εξερχόμενα')
   while AP+es-ek <0 or AP+es-ek >170 :
      es=input('Δωσε ΣΩΣΤΑ εισερχόμενα')
ek=input('Δωσε ΣΩΣΤΑ εξερχόμενα')
AP=AP+es-ek

   ES.append(es)
   EK.append(ek)
   A.append(AP)

   D=raw_input("Τέλος Εισαγωγής Στοιχείων; ΝΑΙ / ΟΧΙ")

max1=0
for x in ES:
   ifx>max1:
max1=x

print "μέγιστος ημερήσιος αριθμός εισερχόμενων", max1

S=0.0
for x  in range(len(ES)):
   S=S+ES[x]+EK[x]

print "μέση ημερήσια διακίνηση ", S/len(ES)

c=0
S1=0.0
for x in A:
   if x>=10:
      c=c+1

S1=S1+x

print "Ημερες με τουλάχιστον 10 κιβώτια", c

print "μέσος όρος αποθηκευτικού χώρου ", S1/len(A)
```

ΑΣΚΗΣΗ 2

```
def Y_E():
   x=0
   e=input("Δώσε τιμή ρίπου")
   while e!=0:
      x=x+1
      if x==1:
         max1=e
      if e>max1:
         max1=e
      e=input("Δώσε τιμή ρίπου")

   return max1

ON=[]
EP=[]

N=input("Δώσε αριθμό ποταμών")
while N>20:
   N=input("Δώσε ΣΩΣΤΟ αριθμό ποταμών")

for x in range(N):
   on=str(raw_input("Δώσε ονομα ποταμού"))
   ON.append(on)
   T=[]
   for y in range(12):
      print "ΜΗΝΑΣ",y+1
      T.append(Y_E())
   EP.append(T)

F=False
for x in range(N):
   S=0.0
   for y in range(12):
      S=S+EP[x][y]
   if S/12 >7 :
      print ON[x]
      F=True

if F==False:
   print "δεν υπάρχει ποταμός με επικινδυνότητα > 7"
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2018

1)Μια συνεταιριστική γεωργική μονάδα επεξεργάζεται στο αποστακτήριό της ένα ελληνικό αρωματικό φυτό και παράγει αιθέριο έλαιο. Στο αποστακτήριο εισάγονται δέματα και κάθε δέμα ζυγίζεται. Το βάρος κάθε δέματος εισάγεται σε ένα πληροφοριακό σύστημα. Μετά την απόσταξη κάθε δέματος το αιθέριο έλαιο που παράγεται ζυγίζεται και το βάρος του εισάγεται επίσης στο πληροφοριακό σύστημα.Μετά το τέλος της παραγωγής το αιθέριο έλαιο συσκευάζεται σε φιαλίδια που περιέχουν 2 γραμμάρια προϊόντος το καθένα.

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Γ1.

α. να περιέχει κατάλληλο τμήμα δηλώσεων ,

β. να διαβάζει το βάρος κάθε δέματος σε κιλά και το βάρος του παραγόμενου αιθέριου ελαίου σε γραμμάρια (πραγματικοί αριθμοί). Η εισαγωγή δεδομένων να τερματίζεται όταν στο ερώτημα:

Θα συνεχιστεί η εισαγωγή; ΝΑΙ/ ΟΧΙ

η απάντηση είναι ΟΧΙ ή όταν ως βάρος του παραχθέντος αιθέριου ελαίου δοθεί η τιμή 0.

Γ2. Να υπολογίζει και να εμφανίζει με κατάλληλα μηνύματα το πλήθος των δεμάτων που εισήχθησαν και το συνολικό βάρος του αιθέριου ελαίου που παρήχθη.

Γ3. Να βρίσκει και να εμφανίζει τη σειρά εισαγωγής που είχε το δέμα εκείνο από το οποίο παρήχθη η μεγαλύτερη ποσότητα αιθέριου ελαίου (να θεωρήσετε ότι το δέμα αυτό είναι μοναδικό).

Γ4. Να υπολογίζει και να εμφανίζει τον συνολικό αριθμό φιαλιδίων που γέμισαν .

Γ5. Να υπολογίζει και να εμφανίζει τον μέγιστο αριθμό διαδοχικών δεμάτων από τα οποία παρήχθηη ίδια ποσότητα αιθέριου ελαίου. (Να θεωρήσετε ότι υπάρχουν δύο τουλάχιστον τέτοια διαδοχικά δέματα).

(Να θεωρήσετε ότι δεν απαιτείται έλεγχος εγκυρότητας για τις τιμές εισόδου).

2) Ένα κλιμάκιο της οργάνωσης «Γιατροί της Ελλάδας » επισκέπτεται τους καλοκαιρινούς μήνες 15 απομονωμένα νησιά προσφέροντας ιατρικές υπηρεσίες. Το πρόγραμμα επισκέψεων ολοκληρώνεται όταν το κλιμάκιο επισκεφτεί, τουλάχιστον μία φορά, και τα 15 νησιά ενώ, αν χρειαστεί, μπορεί να επισκεφτεί κάποια νησιά περισσότερες από μία φορές.

Να κατασκευάσετε πρόγραμμα σε Python το οποίο:

Δ1.

α. Να διαβάζει τα ονόματα των νησιών και να τα καταχωρίζει σε λίστα ΟΝ.

β. Να διαβάζει για κάθε νησί την απόσταση του από τον Πειραιά και να καταχωρίζει την τιμή σε λίσταΑ.

Δ2. Υλοποιώντας κατάλληλη επαναληπτική διαδικασία, για καθεμιά από τις μετακινήσεις του κλιμακίου :

α. να διαβάζει τον αριθμό του νησιού (1 έως 15) προς το οποίο θα γίνει η μετακίνηση,

β. να υπολογίζει το πλήθος των επισκέψεων που έγιναν στο νησί αυτό και να το αποθηκεύει στην αντίστοιχη θέση λίστας Εκαι

γ. να τερματίζει την επαναληπτική διαδικασία μόλις ολοκληρωθεί το πρόγραμμα επισκέψεων.

Δ3. Μετά την ολοκλήρωση του προγράμματος επισκέψεων να εμφανίζει:

α. τα ονόματα των νησιών και το πλήθος των επισκέψεων που δέχθηκε το καθένα,

β. τη συνολική απόσταση που διάνυσε το κλιμάκιο (και για να πάει στο νησί και για να επιστρέψει).

ΑΣΚΗΣΗ 1

```
c=0
BE=0
max1=0
pr=-1
c2=1

b=float(input("Δώσε βάρος δέματος κιλά"))

be=float(input("Δώσε βάρος αιθέριου ελαίου"))
E="NAI"

while E!="OXI" and be!=0:

   c=c+1
   BE=BE+be

   if be>max1:
      max1=be
      Z=c

   if be==pr:
      c2=c2+1
   else:
      if c2>max2:
         max2=c2
c2=1

b=float(input("Δώσε βάρος δέματος κιλά"))

be=float(input("Δώσε βάρος αιθέριου ελαίου"))

E=raw_input("Θα συνεχιστεί η εισαγωγή; NAI/ OXI")


print "πλήθος δεμάτων που εισήχθησαν",c

print "συνολικό βάρος αιθέριου ελαίου ",BE

print "σειρά εισαγωγής της μεγαλύτερης ποσότητας αιθέριου ελαίου",Z

print "συνολικός αριθμός φιαλιδίων ", BE//2

print "μέγιστο αρ.διαδοχικών δεμ. με ίδια ποσότητα ελαίου",max2
```

ΑΣΚΗΣΗ 2

```
ON=[]
A=[]
E=[]

for x in range (15):
   on=str(raw_input("Δώσε ονομα νησιού"))
a=input("Δώσε απόσταση απο τον Πειραιά")

ON.append(on)
   A.append(a)

   E.append(0)

S=0

while 0 in E:
   N=input("Δώσε αριθμό νησιού")

E[N-1]+=1

   S=S+A[N-1]*2

for x in range (15):
   print ON[x], E[x]

print "συνολική απόσταση που διάνυσε το κλιμάκιο ", S
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2018 – ΟΜΟΓΕΝΕΙΣ ΕΞΩΤΕΡΙΚΟΥ

1) Το Υπουργείο Υγείας έκανε 12 διαφορετικές μετρήσεις ακτινοβολίας, μία για κάθε μήνα σε διάρκεια ενός έτους, σε καθένα από 20 νοσοκομεία της Αττικής.

Να γραφεί πρόγραμμα σε Python το οποίο :

Δ1. Να διαβάζει:

α. τα ονόματα των νοσοκομείων και να τα καταχωρίζει σε λίστα ΟΝ.

β. τις τιμές όλων των μετρήσεων και να τις καταχωρίζει σε λίστα ΤΙΜΗ.

Δ2. Να υπολογίζει τον μέσο όρο των τιμών των μετρήσεων ακτινοβολίας κάθε νοσοκομείου και να καταχωρίζει τους μέσους όρους που υπολόγισε σε λίστα ΜΟ.

Δ3. Να εμφανίζει:

α. τη μέγιστη τιμή της λίστας ΜΟ.

β. τη λέξη ΝΟΣΟΚΟΜΕΙΟ και δίπλα το όνομα του νοσοκομείου που έχει την παραπάνω μέγιστη τιμή μέσου όρου (εφόσον υπάρχει μόνο ένα τέτοιο νοσοκομείο) ή τη λέξη ΝΟΣΟΚΟΜΕΙΑ και τα ονόματα όλων των νοσοκομείων που έχουν την παραπάνω μέγιστη τιμή του μέσου όρου (εφόσον υπάρχουν περισσότερα από ένα τέτοια νοσοκομεία).

Δ4. Να διαβάζει το όνομα ενός νοσοκομείου και να εμφανίζει το μήνυμα ΔΕΝ ΥΠΑΡΧΕΙ, αν δεν υπάρχει στην λίστα ΟΝ το συγκεκριμένο νοσοκομείο, ή το πλήθος των τιμών μέτρησης ακτινοβολίας του νοσοκομείου που είναι μεγαλύτερες του μέσου όρου του νοσοκομείου, αν υπάρχει στην λίστα ΟΝ το συγκεκριμένο νοσοκομείο .

(Να θεωρήσετε ότι δεν απαιτείται έλεγχος εγκυρότητας για τις τιμές εισόδου).

ΑΣΚΗΣΗ 1

```
ON=[] ;  TINMH=[] ;  MO=[] ;  A=[]

for x in range(20):
   on=str(raw_input("Δώσε όνομα"))
   ON.append(on)

   T=[]
   for y in range(12):
      print "ΜΗΝΑΣ",y+1
      m=input("Δώσε τιμή")
      T.append(m)
   TIMH.append(T)

for x in range(20):
   S=0.0
   for y in range(12):
      S=S+TIMH[x][y]

   MO.append(S/12)

max1=0
for x in range(20):
   if MO[x]>max1:
      max1=MO[x]

for x in range(20):
   if MO[x]==max1:
      A.append(ON[x])


if len(A)==1:
   print "ΝΟΣΟΚΟΜΕΙΟ"
   print A[0]
else:
   print "ΝΟΣΟΚΟΜΕΙΑ"
   for x in A:
      print x

c=0
onoma=raw_input("Δώσε όνομα")

if onoma not in  ON:
   print "ΔΕΝ ΥΠΑΡΧΕΙ"
else:
   for i in range(20):
      if onoma==ON[x]:
         Z=x

      for y in range(12):
         if TIMH[Z][y]>MO[x]:
            c=c+1
print "πλήθος των τιμών μέτρησης μεγαλύτερες του μέσου όρου",c
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2019

1) Το Υπουργείο Παιδείας παρέχει μέσω του διαδικτύου μια συλλογή από εκπαιδευτικά βίντεο. Ο αριθμός των επισκέψεων που δέχεται κάθε ένα βίντεο καταγράφεται από ειδικό λογισμικό. Τα βίντεο διακρίνονται σε τρείς κατηγορίες ανάλογα με την επισκεψιμότητά τους, σύμφωνα με τον παρακάτω πίνακα:

ΚΑΤΗΓΟΡΙΕΣ ΕΠΙΣΚΕΨΙΜΟΤΗΤΑΣ	
Όνομα	**Αριθμός Επισκέψεων**
Χαμηλή	από 1 έως και 100
Μεσαία	από 101 έως και 1000
Υψηλή	πάνω από 1000

Τα βίντεο με μηδενικές επισκέψεις δεν κατατάσσονται σε καμία κατηγορία.

Να αναπτύξετε πρόγραμμα σεPython το οποίο:

Γ1. Να διαβάζει επαναληπτικά τον τίτλο κάθε βίντεο και τον αριθμό των επισκέψεων που δέχτηκε. Η είσοδος των δεδομένων να τερματίζεται, όταν ως τίτλος βίντεο δοθεί η λέξη «ΤΕΛΟΣ».

Να γίνεται έλεγχος εγκυρότητας ώστε ο αριθμός των επισκέψεων να μην είναι αρνητικός.

Γ2. Να βρίσκει και να εμφανίζει τον τίτλο του βίντεο με τον μεγαλύτερο αριθμό επισκέψεων. Να θεωρήσετε ότι είναι μοναδικό.

Γ3. Να υπολογίζει για καθεμία από τις τρεις κατηγορίες επισκεψιμότητας το πλήθος των βίντεο που καταχωρίστηκαν σε αυτή . Να εμφανίζει για κάθε κατηγορία:

- το όνομά της και
- το πλήθος των βίντεο που περιλαμβάνει.

Γ4. Να βρίσκει και να εμφανίζει το όνομα της κατηγορίας επισκεψιμότητας στην οποία καταχωρίστηκαν τα περισσότερα βίντεο. Να θεωρήσετε ότι είναι μοναδική.

Σημείωση

Το πλήθος των βίντεο δεν είναι γνωστό.

2) Στην 27η Βαλκανιάδα Πληροφορικής που θα διεξαχθεί στην Αθήνα τον Σεπτέμβριο του 2019, συμμετέχουν 40 μαθητές. Κάθε μαθητής παίρνει έναν κωδικό **από 1 έως και 40**, ο οποίος αντιστοιχεί στη σειρά που δήλωσε συμμετοχή. Κάθε μαθητής καλείται να επιλύσει **τέσσερα προβλήματα**. Για κάθε πρόβλημα αναπτύσσει τη λύση του σε μία γλώσσα προγραμματισμού και την υποβάλλει για βαθμολόγηση. Η λύση βαθμολογείται σε ακέραια κλίμακα από 0 έως 100.

Κατά τη διάρκεια του διαγωνισμού κάθε μαθητής και για κάθε πρόβλημα μπορεί να υποβάλλει τη λύση του όσες φορές θέλει .

Να αναπτύξετε πρόγραμμα σε Python το ο ποίο:

Δ1. Να διαβάζει επαναληπτικά τα ονόματα των μαθητών και να τα καταχωρίζει στην λίστα ΟΝ.

Επίσης, να αρχικοποιεί με την τιμή 0 όλα τα στοιχεία των λιστών V1, V2, V3, V4, οι οποίες θα περιέχει τη βαθμολογία κάθε μαθητή για κάθε πρόβλημα.

Δ2. Κάθε φορά που μία λύση προβλήματος υποβάλλεται και βαθμολογείται, το πρόγραμμα να διαβάζει τον κωδικό του μαθητή (από 1 έως και 40), τον αριθμό του προβλήματος (από 1 έως και 4) και τη βαθμολογία του (από 0 έως και 100) .

Η βαθμολογία να καταχωρίζεται στην αντίστοιχη θέση των λιστών V1, V2, V3, V4 μόνο αν είναι μεγαλύτερη από τη αντίστοιχη βαθμολογία που είναι ήδη καταχωρισμένη .

Για τον τερματισμό της εισαγωγής δεδομένων το πρόγραμμα να εμφανίζει το μήνυμα «Υπάρχει νέα λύση προβλήματος; ΝΑΙ / ΟΧΙ». Αν εισαχθεί η τιμή «ΟΧΙ», να τερματίζεται η εισαγωγή δεδομένων.

Δ3. Να υπολογίζει και να καταχωρίζει στην λίστα SUM τα αθροίσματα των βαθμολογιών κάθε μαθητή στα 4 προβλήματα. Για τον σκοπό αυτό να καλεί μόνο μια φορά το υποπρόγραμμα με όνομα YSB.

Να αναπτύξετε το υποπρόγραμμα YSB το οποίο να δέχεται ως είσοδο τις λίστες V1, V2, V3, V4 και να επιστρέφει ως έξοδο συμπληρωμένη την λίστα SUM .

Δ4. Να εμφανίζει τα ονόματα των μαθητών ταξινομημένων σύμφωνα με τη συνολική τους βαθμολογία σε φθίνουσα βαθμολογική σειρά. Σε περίπτωση μαθητών με την ίδια βαθμολογία, τα ονόματά τους να εμφανίζονται με αλφαβητική σειρά.

Σημειώσεις

α) Δεν απαιτούνται έλεγχοι εγκυρότητας τιμών .

β) Να θεωρήσετε ότι θα δοθεί τουλάχιστον μια λύση προβλήματος από έναν μαθητή .

ΑΣΚΗΣΗ 1

```
c1=0
c2=0
c3=0
max1=0

T=raw_input("Δώσε τίτλο")

while T!="ΤΕΛΟΣ":
   E=input("Δώσε αριθμό επισκέψεων")
   while E<0:
      E=input("Δώσε ΣΩΣΤΟ αριθμό επισκέψεων")

   if E>max1:
      max1=E
      Tm=T

   if E>=1 and E<=100:
      c1=c1+1
   if E>=101 and E<=1000:
      c2=c2+1
   if E>1000:
      c3=c3+1

   T=raw_input("Δώσε τίτλο")

print "τίτλο με τον μεγαλύτερο αριθμό επισκέψεων",Tm

print "Χαμηλή",c1

print "Μεσαία",c2

print "Υψηλή",c3

max2=c1
k="Χαμηλή"
if c2>max2:
   max2=c2
   k="Μεσαία"
if c3>max2:
   max2=c3
   k="Υψηλή"

print "κατηγορία με μεγαλύτερη επισκεψιμότητα",k
```

ΑΣΚΗΣΗ 2

```
def YSB(V1,V2,V3,V4):
   S=[]
   for x in range(40):
      S.append(V1[x]+V2[x]+V3[x]+V4[x])

   return S

ON=[]
V1=[] ; V2=[] ; V3=[] ; V4=[]
SUM=[]

for x in range(40):
   on=str(raw_input("Δώσε όνομα"))
   ON.append(on)

   V1.append(0)
   V2.append(0)
   V3.append(0)
   V4.append(0)

E="NAI"

while E=="NAI":

   K=input("Δώσε κωδικό μαθητή")
   K=K-1

   A=input("Δώσε αριθμό του προβλήματος")

   V=input("Δώσε βαθμολογία")

   if A==1 and V1[K]<V:
      V1[K]=V
   if A==2 and V2[K]<V:
      V2[K]=V
   if A==3 and V3[K]<V:
      V3[K]=V
   if A==4 and V4[K]<V:
      V4[K]=V

   E=raw_input("Υπάρχει νέα λύση
προβλήματος; NAI / OXI")

SUM=YSB(V1,V2,V3,V4)
```

```
N=40

for i in range(1,N,1):

   for j in range(N-1,i-1,-1):

      if  SUM[j]<SUM[j-1]:
         SUM[j],SUM[j-1]=SUM[j-1],SUM[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         V1[j],V1[j-1]=V1[j-1],V1[j]
         V2[j],V2[j-1]=V2[j-1],V2[j]
         V3[j],V3[j-1]=V3[j-1],V3[j]
         V4[j],V4[j-1]=V4[j-1],V4[j]

      if  SUM[j]==SUM[j-1]  and  ON[j]<ON[j-1]:
         SUM[j],SUM[j-1]=SUM[j-1],SUM[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         V1[j],V1[j-1]=V1[j-1],V1[j]
         V2[j],V2[j-1]=V2[j-1],V2[j]
         V3[j],V3[j-1]=V3[j-1],V3[j]
         V4[j],V4[j-1]=V4[j-1],V4[j]

for x in ON:
   print  x
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΕΣΠΕΡΙΝΟΥ ΓΕΛ 2019

1)Το Υπουργείο Παιδείας μελετά το πλήθος των αγοριών και των κοριτσιών που φοιτούν σε κάθε τμήμα της Γ΄ τάξης μιας ομάδας λυκείων, για στατιστικούς λόγους.

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Γ1. Να διαβάζει :

- για κάθε λύκειο, το όνομά του, το πλήθος των τμημάτων της Γ΄ τάξης και
- για κάθε τμήμα της Γ΄ τάξης κάθε λυκείου , το πλήθος των αγοριών και των κοριτσιών.

Η εισαγωγή των δεδομένων να τερματίζεται, όταν δοθεί, ως όνομα λυκείου, η λέξη "ΤΕΛΟΣ".

Να θεωρήσετε ότι υπάρχει ένα τουλάχιστον λύκειο και κάθε λύκειο έχει ένα τουλάχιστον τμήμα.

Γ2. Να υπολογίζει και να εμφανίζει για κάθε λύκειο, το συνολικό πλήθος των μαθητών της Γ΄ τάξης, τον μέσο όρο των μαθητών ανά τμήμα και το πλήθος των ολιγομελών τμημάτων, δηλαδή των τμημάτων με λιγότερους από 15 μαθητές.

Γ3. Να υπολογίζει για κάθε λύκειο, το πλήθος των τμημάτων της Γ΄ τάξης στα οποία τα κορίτσια είναι περισσότερα από τα αγόρια και να εμφανίζει ένα από τα παρακάτω:

α) το μήνυμα " ΤΑ ΚΟΡΙΤΣΙΑ ΕΙΝΑΙ ΠΕΡΙΣΣΟΤΕΡΑ ΣΕ ΟΛΑ ΤΑ ΤΜΗΜΑΤΑ "

β) το μήνυμα " ΔΕΝ ΥΠΑΡΧΕΙ ΤΜΗΜΑ ΟΠΟΥ ΤΑ ΚΟΡΙΤΣΙΑ ΕΙΝΑΙ ΠΕΡΙΣΣΟΤΕΡΑ ΑΠΟ ΤΑ ΑΓΟΡΙΑ "

γ) το πλήθος των τμημάτων στα οποία τα κορίτσια είναι περισσότερα από τα αγόρια, εφόσον δεν ισχύει κάποια από τις περιπτώσεις α ή β .

Γ4.Να εντοπίζει και να εμφανίζει το όνομα του λυκείου με τον μέγιστο συνολικό αριθμό κοριτσιών στη Γ΄ τάξη (να θεωρήσετε ότι το λύκειο αυτό είναι μοναδικό).

2) Σε ένα μουσικό φεστιβάλ συμμετέχουν 20 συγκροτήματα. Τα ονόματά τους καταχωρίζονται σε λίστα ΟΝ.

Το φεστιβάλ διαρκεί 5 ημέρες και κάθε ημέρα εμφανίζονται 6 συγκροτήματα. Το πρόγραμμα εμφανίσεων των συγκροτημάτων περιγράφεται με μία λίσταΡ. Σε κάθε θέση της λίστας Ρ αντιστοιχεί στις ημέρες του φεστιβάλ και καταχωρίζονται υπολίστεςπου κάθε μία περιέχειτιςαντίστοιχες θέσεις του συγκροτήματος στηνλίστα ΟΝ.

Κάποια συγκροτήματα εμφανίζονται σε περισσότερες από μια ημέρες και κανένα δεν εμφανίζεται περισσότερες από μία φορά την ημέρα.

Να αναπτύξετε πρόγραμμα το οποίο:

Δ1. Να διαβάζει τα ονόματα των συγκροτημάτων και να τα καταχωρίζει στην λίστα ΟΝ.

Δ2. Για κάθε μία από τις 5 η μέρες, να διαβάζει τους αριθμούς των 6 συγκροτημάτων που εμφανίζονται την ημέρα αυτή, με τη σειρά που εμφανίζονται, και να τους καταχωρίζει στις αντίστοιχες θέσεις της λίστας Ρ. Κάθε τιμή που εισάγεται να γίνεται δεκτή μόνο εάν δεν έχει ξαναεισαχθεί την ίδια ημέρα, διαφορετικά να ζητείται ξανά. Ο έλεγχος αυτός να γίνεται από το υποπρόγραμμα ΥΠΑΡΧΕΙ.

Δ3. Για καθένα από τα 20 συγκροτήματα να τυπώνει το όνομά του και το πρόγραμμα εμφανίσεών του, δηλαδή μόνο τις ημέρες που εμφανίζεται και για κάθε μία από αυτές τη σειρά εμφάνισής του.

Δ4. Να τυπώνει τοόνομα τουσυγκροτήματος που εμφανίζονται τις περισσότερες φορές.

Σημείωση: Να θεωρήσετε ότι δεν απαιτούνται επιπλέον έλεγχοι εγκυρότητας για τις τιμές εισόδου.

ΑΣΚΗΣΗ 1

```
ON=[] ;  P=[] ;  K=[] ;  A=[]
max1=0

on=str(raw+input("Δώσε όνομα Λυκείου"))
while on!="ΤΕΛΟΣ":
   n=input("Δώσε πλήθος τμημάτων Γ΄ τάξης")
   ON.append(on)
   P.append(n)

   TK=[]
   TA=[]
   for x in range(n):
      print "Τμήμα:", x+1
k=input("Δώσε πλήθος κοριτσιών")
a=input("Δ¨ωσε πλήθος αγοριών")
TK.append(k)
      TA.append(a)
   K.append(TK)
   A.append(TA)

   on=str(raw+input("Δώσε όνομα Λυκείου"))

for x in range(len(ON)):
   S=0.0
   c=0
   z=0
   SW=0
   for y in range(P[x]):
      S=S+K[x][y] + A[x][y]
      if K[x][y] + A[x][y] <15:
         c=c+1

      if K[x][y]>A[x][y]:
         z=z+1

      SW=SW+K[x][y]

   print "Σϋνολο μαθητών Γ΄ τάξης",S
   print "Μέσος όρος", S/P[x]
print "Πλήθος ολιγομελών τμημάτων",c

ifz==0:
print "ΔΕΝ ΥΠΑΡΧΕΙ ΤΜΗΜΑ ΟΠΟΥ ΚΟΡΙΤΣΙΑ > ΑΓΟΡΙΑ"
ifz==P[x]:
print "ΤΑ ΚΟΡΙΤΣΙΑ ΕΙΝΑΙ ΠΕΡΙΣΣΟΤΕΡΑ ΣΕ ΟΛΑ ΤΑ ΤΜΗΜΑΤΑ "
ifx<P[x]:
print "πλήθος τμημάτων όπου κορίτσια  > αγόρια", z

if SW>max1:
      max1=SW
      onoma=ON[x]
print "Λύκεο με τον μέγιστο συνολικό αριθμό κοριτσιών",onoma
```

ΑΣΚΗΣΗ 2

```
def yparxei(T,a):
   if a in T:
      return True
   else:
      return False


ON=[]
P=[]


for x in range (20):
   on=str(raw_input("Δώσε όνομα συγκροτήματος"))
ON.append(on)

for x in range(5):
   T=[]
   for y in range(6):

      while yparxei(T,a)==True:
         a=input("Δώσε σωστο αριθμό συγκροτήματος")
T.append(a)
   P.append(T)

max1=0
for x in range (20):
   c=0
   print ON[x]
   for y in range(5):
      for z in range(6):

         if P[y][z] ==x:
            print "ημέρα που εμφανίζεται", y
c=c+1
if c>max1:
      max1=c
      Z=x

print "εμφανίζονται τις περισσότερες φορές",ON[Z]
```

ΠΑΝΕΛΛΗΝΙΕΣ ΕΞΕΤΑΣΕΙΣ ΓΕΛ 2020

1) Ένα πλοίο μεταφέρει δέματα από λιμάνια της Ελλάδας στην Ιταλία. Σε κάθε λιμάνι που καταπλέει για φόρτωση δηλώνει το βάρος που έχει ήδη φορτωμένο, καθώς και το μέγιστο βάρος που μπορεί να μεταφέρει (όριο βάρους). Η διαδικασία φόρτωσης ελέγχεται από αρμόδιο υπάλληλο.

Να αναπτύξετε πρόγραμμα σε Python το οποίο να υποστηρίζει τη διαδικασία φόρτωσης σε ένα λιμάνι . Το πρόγραμμα:

Γ1. Να διαβάζει:

- το όριο βάρους του πλοίου,
- το βάρος δεμάτων που έχει ήδη φορτωμένα, ελέγχοντας ότι η τιμή του είναι μικρότερη από το όριο βάρους, διαφορετικά να το ξαναζητά

Γ2. Για τη διαδικασία φόρτωσης:

α) να εμφανίζει το βάρος που μπορεί ακόμα να φορτωθεί στο πλοίο, να εμφανίζει το μήνυμα: «ΝΑ ΦΟΡΤΩΘΕΙ ΔΕΜΑ; (ΝΑΙ/ΟΧ Ι)» , να διαβάζει την απάντηση του αρμόδιου υπαλλήλου (χωρίς έλεγχο εγκυρότητας) .

β) Αν η απάντηση είναι «ΝΑΙ» να διαβάζει το βάρος του δέματος,

- να ελέγχει ότι δεν παραβιάζεται το όριο βάρους και να επιτρέπει τη φόρτωσή του, διαφορετικά να εμφανίζει το μήνυμα «ΤΟ ΔΕΜΑ ΔΕΝ ΧΩΡΑΕΙ» , εφόσον επιτραπεί η φόρτωσή του ,
- να υπολογίζει και να εμφανίζει το κόστος μεταφοράς του **κλιμακωτά**, με βάση το βάρος του, ως εξής:
  - τα πρώτα 500 κιλά χρεώνονται 0,5 € / κιλό,
  - τα επόμενα 1000 κιλά χρεώνονται 0,3 € / κιλό,
  - τα υπόλοιπα χρεώνονται 0,1 € / κιλό.

Η παραπάνω διαδικασία φόρτωσης επαναλαμβάνεται μέχρι να δοθεί ως απάντηση από τον αρμόδιο υπάλληλο η λέξη «ΟΧΙ».

Γ3.Μετά το τέλος φόρτωσης να εμφανίζει:

- πόσα από τα δέματα που ελέγχθηκαν δεν φορτώθηκαν λόγω υπέρβασης του ορίου βάρους,
- το συνολικό ποσό που εισπράχθηκε,
- το πλήθος των δεμάτων που φορτώθηκαν και είχαν βάρος που ξεπερνούσε τα 1000 κιλά

2) Οι Κινητές Ομάδες Υγείας (ΚΟΜΥ) λαμβάνουν δείγματα βιολογικού υλικού προσώπων για έλεγχο μόλυνσης από τον κορωνοϊό Covid- 19. Σε μια περιφέρεια δραστηριοποιούνται 20 ΚΟΜΥ. Κάθε ΚΟΜΥ στη διάρκεια μιας μέρας μπορεί να λάβει μέχρι και 100 δείγματα από μια περιοχή της περιφέρειας . Τα δείγματα αυτά ελέγχονται και κάθε αποτέλεσμα χαρακτηρίζεται ως θετικό (Θ) ή αρνητικό (Α) και καταγράφεται σε πληροφοριακό σύστημα.

Να αναπτύξετε πρόγραμμα σε Python το οποίο:

Δ1. α) Να διαβάζει τα ονόματα των περιοχών που δραστηριοποιούνται οι ΚΟΜΥ και να τα καταχωρίζει σε λίστα ΟΝ.

β) Για κάθε ΚΟΜΥ να διαβάζει διαδοχικά τα αποτελέσμα τα των ελέγχων που έχει πραγματοποιήσει και κάθε αποτέλεσμα να το καταχωρίζει ως ένα γράμμα Α ή Θ στην αντίστοιχη θέση της λίστας Α. Σε περίπτωση που λήφθηκαν λιγότερα από 100 δείγματα, μετά την καταχώριση του αποτελέσματος του τελευταίου δείγματος διαβάζεται αντί αποτελέσματος η λέξη «ΤΕΛΟΣ», η οποία δεν καταχωρίζεται στην λίστα Α. Σε αυτή την περίπτωση τερματίζεται η εισαγωγή τιμών για τη συγκεκριμένη ΚΟΜΥ και το πρόγραμμα καταχωρίζει σε όλες τις υπόλοιπες θέσεις το γράμμα Χ.

Δ2. Να εμφανίζει το όνομα ή τα ονόματα των περιοχών που βρέθηκαν τα περισσότερα θετικά δείγματα.

Δ3. Να εμφανίζει τα ονόματα των περιοχών, ταξινομημένα σε φθίνουσα σειρά ως προς το πλήθος των θετικών δειγμάτων που εντοπίστηκαν. Σε περίπτωση που δύο ή περισσότερες περιοχές έχουν το ίδιο πλήθος θετικών δειγμάτων, τα ονόματά τους να εμφανίζονται με αλφαβητική σειρά. Για την ταξινόμηση να καλείται το υποπρόγραμμα ΤΑΞΙΝΟΜΗΣΗ του ερωτήματος Δ4.

Δ4.Να αναπτύξετε υποπρόγραμμα με όνομα ΤΑΞΙΝΟΜΗΣΗ, που υλοποιεί τη λειτουργία της ταξινόμησης που περιγράφεται στο ερώτημα Δ3.

Σημειώσεις

- Για την απάντηση των ερωτημάτων Δ2, Δ3 και Δ4 να θεωρήσετε ότι η λίστα Α έχει συμπληρωθεί σωστά.

- Δεν απαιτούνται έλεγχοι εγκυρότητας τιμών .

- Να θεωρήσετε ότι τα ονόματα των περιοχών είναι διαφορετικά μεταξύ τους.

ΑΣΚΗΣΗ 1

```
SXr=0
c=0
c1=0

ORIO=input("Δώσε όριο βάρους")

B=input("Δώσε βάρος δεμάτων που έχει ήδη φορτωμένα")

print "βάρος που μπορεί ακόμα να φορτωθεί ",ORIO-B
E=raw_input("ΝΑ ΦΟΡΤΩΘΕΙ ΔΕΜΑ; (ΝΑΙ/ΟΧΙ)")

while E!="ΟΧΙ":

   B1=input("Δώσε βάρος δέματος")
   if B1>ORIO-B:
      print "ΤΟ ΔΕΜΑ ΔΕΝ ΧΩΡΑΕΙ"
      c1=c1+1
   else:
      B=B+B1
      if B1>=1 and B1<=500:
         Xr=B1*0.5
      if B1>=501 and B1<=1500:
         Xr=500*0.5+(B1-500)*0.3
      if B1>1500:
         Xr=500*0.5+ 1000*0.3+(B1-1500)*0.10

      SXr=SXr+Xr

      if B1>1000:
         c=c+1


   print "βάρος που μπορεί ακόμα να φορτωθεί ",ORIO-B
   E=raw_input("ΝΑ ΦΟΡΤΩΘΕΙ ΔΕΜΑ; (ΝΑΙ/ΟΧΙ)")

print "δέματα που δεν φορτώθηκαν ", c1
print "συνολικό ποσό που εισπράχθηκε",SXr
print "δέματα με βάρος που ξεπερνούσε τα 1000 κιλά",c
```

ΑΣΚΗΣΗ 2

```
def TAXINOMHSH(ON,A,B):
   N=20
   for i in range(1,N,1):

      for j in range(N-1,i-1,-1):

         if  B[j]>B[j-1]:
            B[j],B[j-1]=B[j-1],B[j]
            ON[j],ON[j-1]=ON[j-1],ON[j]
            A[j],A[j-1]=A[j-1],A[j]

         if  B[j]==B[j-1]  and  ON[j]<ON[j-1]:
            B[j],B[j-1]=B[j-1],B[j]
            ON[j],ON[j-1]=ON[j-1],ON[j]
            A[j],A[j-1]=A[j-1],A[j]

ON=[]  ;  A=[]  ;  B=[]

for x in range(20):
   on=str(raw_input("Δώσε όνομα περιοχής"))
   ON.append(on)

   T=[]
   F=False
   for y in range(100):

      if F==False:
         m=str(raw_input("Δώσε δείγμα"))
         if m=="ΤΕΛΟΣ":
            F=True
         else:
            T.append(m)
      if F==True:
         T.append("X")
   A.append(T)

max1=0
for x in range(20):
   S=0
   for y in range(100):
      if A[x][y]=="Θ":
         S=S+1
   B.append(S)
   if S>max1:
      max1=S
      onoma=ON[x]

print "περιοχή με τα περισσότερα θετικά δείγματα",onoma
TAXINOMHSH(ON,A,B)
for x in ON:
   print x
```

ΠΑΡΑΡΤΗΜΑ

Τεχνικές για Λίστες

Η συνάρτηση range(A, M, B) επιστρέφει μια λίστα αριθμών ξεκινώντας με τον αριθμό Α μέχρι το Μ με βήμα Β. Το Μ δεν συμπεριλαμβάνεται στη λίστα.

Οι παρακάτω κώδικες έχουν ακριβώς τα ίδια αποτελέσματα:

Α)

```
L = [ 6, 28, 496, 81, 28 ]

for   x   in    L :

          print   x
```

Β)

```
L = [ 6, 28, 496, 81, 28 ]

for    x   in   range(len(L)) :

          print    L[x]
```

Γ)

```
L = [ 6, 28, 496, 81, 28 ]

for    x   in   [ 0, 1, 2, 3, 4 ] :

          print    L[x]
```

Επιλογή στοιχείων που θα τυπωθούν:

Α)

```
L = [ 6, 28, 4, 9, 15, 81, 28 ]

for    x   in   range(0, len(L), 2) :

          print  L[x]                τυπώνει : 6, 4, 15, 28   τυπώνει ανά δύο
```

Β)

```
L = [ 6, 28, 4, 9, 15, 81, 28 ]

for    x   in   L[  : len(L) -2] :

          print  x                   τυπώνει : 6, 28, 4, 9, 15 δεν τυπώνει τα δύο τελευταία
```

Γ)

```
L = [ 6, 28, 4, 9, 15, 81, 28 ]
for   x  in  L[  2 :   ] :
        print   x
```

τυπώνει : 4, 9, 15, 81, 28 ***δεν τυπώνει τα δύο πρώτα***

Δ)

```
L = [ 6, 28, 4, 9, 15, 81, 28 ]
for   x  in   L :
        if   x > 10 :
              print  x
```

τυπώνει : 28, 15, 81, 28 **τυπώνει υπό συνθήκη** δηλαδή όσα είναι μεγαλύτερα από 10

Διαβάζω και τυπώνω τα στοιχεία μίας λίστας και ταυτόχρονα την αδειάζω από την αρχή

```
L = [ 6, 28, 4, 9, 15, 81, 28 ]
while   L != []:
        print   L[0]
        L.pop(0)
```

Τυπώνω τα στοιχεία μιας λίστας από το τέλος προς την αρχή

```
L = [ 6, 28, 4, 9, 15, 81, 28 ]
for   x  in   range(len(L)-1, -1,-1) :
        print   L[x]
```

τυπώνει: 28, 81, 15, 9, 4, 28, 6

Δημιουργώ την αντίστροφη μιας λίστας

L = [6, 28, 4, 9, 15, 81, 28] L1=[] for x in **range(len(L)-1, -1,-1)** : L1.append(L[x])	L = [6, 28, 4, 9, 15, 81, 28] L1=[] for x in **L**: **L1.insert(0,x)**
L = [6, 28, 4, 9, 15, 81, 28] L1=[28, 81, 15, 9, 4, 28, 6]	L = [6, 28, 4, 9, 15, 81, 28] L1=[28, 81, 15, 9, 4, 28, 6]

Υπολογισμός του **αθροίσματος** των στοιχείων μιας λίστας	Υπολογισμός του **Μέσου Όρου** των στοιχείων μιας λίστας

S=0
L = [6, 28, 4, 9, 15, 81, 28]
for x in **L**:
  S=S+x
	S=0.0
	L = [6, 28, 4, 9, 15, 81, 28]
	for x in **L**:
	  S=S+x
	Mesos_Oros=S/len(L)

Εύρεση του **μεγαλύτερου** στοιχείου μιας λίστας	Εύρεση του **μικρότερου** στοιχείου μιας λίστας

L = [6, 28, 4, 9, 15, **81**, 28]
max1=L[0]
 	
for x in **range(1, len(L))**:
  if L[x] > max1 :
  max1=L[x]
	L = [6, 28, **4**, 9, 15, 81, 28]
	min1=L[0]
	
	for x in **range(1, len(L))**:
	  if L[x] < min1 :
	  min1=L[x]

Λύση Βιβλίου
L = [6, 28, 4, 9, 15, **81**, 28]
max1=L[0]
 	
for x in **L**:
  if x > max1 :
  max1 = x
	Λύση Βιβλίου
	L = [6, 28, **4**, 9, 15, 81, 28]
	min1=L[0]
	
	for x in **L**:
	  if x < min1 :
	  min1 = x

Έχω **δύο λίστες** Α με ονόματα και Β με βαθμολογίες. Θέλουμε να τυπώσουμε το όνομα που αντιστοιχεί στην μεγαλύτερη βαθμολογία

A= [14, 18, 12, 17, 19, 15]
B=[‘Nikos’, ‘Maria’, ‘Anna’, ‘Aris’, ‘Petros’, ‘Tom’]
max1=A[0]
onoma=B[0]
for x in **range(1, len(A))**:
  if A[x] > max1 :
  max1=A[x]
  onoma=B[x]
print **onoma**
	A= [14, 18, 12, 17, 19, 15]
	B=[‘Nikos’, ‘Maria’, ‘Anna’, ‘Aris’, ‘Petros’, ‘Tom’]
	max1=A[0]
	y=0
	for x in **range(1, len(A))**:
	  if A[x] > max1 :
	  max1=A[x]
	  y= x
	print **B[y]**

Έχω **δύο λίστες** Α με ονόματα και Β με βαθμολογίες. Θέλουμε να τυπώσουμε τα ονόματα που αντιστοιχούν στις **τρείς μεγαλύτερες βαθμολογίες**

1ος Τροπος

```
A=[ 14, 18, 12, 17, 19, 15 ]
B=['Nikos', 'Maria', 'Anna', 'Aris', 'Petros', 'Tom']
max1=A[0]
max2=None
max3=None
onoma1=B[0]
onoma2=""
onoma3=""
for   x   in   range(1, len(A)):

   if   A[x] > max1 :
      max3=max2
      max2=max1
      max1=A[x]
      onoma3=onoma2
      onoma2=onoma1
      onoma1=B[x]

   elif   max2==None or A[x] > max2 :
      max3=max2
      max2=A[x]
      onoma3=onoma2
      onoma2=B[x]

   elif  max3==None or A[x] > max3 :
      max3=A[x]
      onoma3=B[x]

print  onoma1, onoma2, onoma3
```

2ος Τροπος

```
A= [ 14, 18, 12, 17, 19, 15 ]
B=['Nikos', 'Maria', 'Anna', 'Aris', 'Petros', 'Tom']

N=len(A)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] > A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]


print  B[0], B[1], B[2]
```

Αν το μεγαλύτερο στοιχείο δεν είναι μοναδικό

```
A= [ 14, 18, 12, 17, 19, 15, 19 ]
B=['Nikos', 'Maria', 'Anna', 'Aris', 'Petros', 'Tom', 'Eleni']
max1=A[0]
onoma=B[0]
for   x   in   range(1, len(A)):
          if   A[x] > max1 :
               max1=A[x]

for   x   in   range(  len(A) ):
         if  A[x] == max1:
              print  B[x]
```

Χρήση μετρητή σε λίστα

Μετράμε πόσα στοιχεία είναι μεγαλύτερα από 15

A= [14, **18**, 12, **17**, **19**, 15, **19**]

c=0

```
for   x   in  A :

        if   x  >  15:

                c=c+1
```

Διαγράφω από μια λίστα τα στοιχεία που ικανοποιούν μια συνθήκη

Διαγράφω τα στοιχεία που είναι μεγαλύτερα από 15

A= [14, **18**, 12, **17**, **19**, 15, **19**]

```
B=[]

for   x   in  A :

        if   x  <=  15:

                B.append(x)

 A=B
```

Διαχωρισμός λίστας

```
B1=[]
B2=[]
A= [ 15, -3, 0, 12, -11, -6, 19, 17 ]
for   x   in  A :
          if   x  >=  0:
                    B1.append(x)
          else :
                    B2.append(x)
```

```
B1=[15, 0, 12, 19, 17]
B2=[-3,  -11, -6]
```

Συγχώνευσης των στοιχείων δυο ταξινομημένων λιστών σε μία νέα, επίσης ταξινομημένη, λίστα

```
A=[14, 22, 35, 38, 66]
B=[8, 12, 16, 39, 45, 72, 80]
L = [ ]
while   A != [ ] and   B != [ ] :
          if   A[0] < B[0] :
                    L.append( A.pop(0) )
          else:
                    L.append( B.pop(0) )
L=L + A + B
```

```
A=[66, 38, 35, 22, 14]
B=[80, 72, 45, 39, 16, 12, 8]
L = [ ]
while   A != [ ] and   B != [ ] :
          if   A[0] > B[0] :
                    L.append( A.pop(0) )
          else:
                    L.append( B.pop(0) )
L=L + A + B
```

φθίνουσα Ταξινόμηση ευθείας ανταλλαγής (φυσαλίδα) για μια λίστα

```
N=len(A)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] > A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
```

Αύξουσα Ταξινόμηση ευθείας ανταλλαγής (φυσαλίδα) για μια λίστα

```
N=len(A)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] < A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
```

φθίνουσα Ταξινόμηση ευθείας ανταλλαγής (φυσαλίδα) για δύο λίστες

```
N=len(A)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] > A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]
```

Αύξουσα Ταξινόμηση ευθείας ανταλλαγής (φυσαλίδα) για δύο λίστες

```
N=len(A)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if A[j] < A[j-1]:
         A[j],A[j-1]=A[j-1],A[j]
         B[j],B[j-1]=B[j-1],B[j]
```

Συνάρτηση φυσαλίδας

```
def   bubbleSort(A):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]

bubbleSort(A)
```

Συνάρτηση φυσαλίδας με RETURN

```
def   bubbleSort(A):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
   return A

A=bubbleSort(A)
```

Συνάρτηση φυσαλίδας για δύο λίστες

```
def   bubbleSort(A,B):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]

bubbleSort(A, B)
```

Συνάρτηση φυσαλίδας ΜΕ RETURN για δύο λίστες

```
def   bubbleSort(A,B):
   N=len(A)
   for i in range(N-1):
      for j in range(N-1,i,-1):
         if A[j]<A[j-1]:
            A[j],A[j-1]=A[j-1],A[j]
            B[j],B[j-1]=B[j-1],B[j]
    return A, B

A, B = bubbleSort(A, B)
```

Ταξινόμηση ως προς το όνομα αν έχουμε συνωνυμία στο επώνυμο

EP=[“Nikolaou”, “**Tsapalis**”, “Zagas”, “**Tsapalis**”]
ON=[“Maria”, “**Petros**”, “Kostas”, “**Giannis**”]
V=[18, **15**, 19, **16**]

Α΄ τρόπος

```
N=len(EP)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if   EP[j]<EP[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         V[j],V[j-1]=V[j-1],V[j]

N=len(EP)
for i in range(1,N,1):
   for j in range(N-1,i-1,-1):
      if   EP[j]==EP[j-1]   and   ON[j]<ON[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         V[j],V[j-1]=V[j-1],V[j]
```

Β΄ τρόπος

```
N=len(EP)

for i in range(1,N,1):

   for j in range(N-1,i-1,-1):

      if   EP[j]<EP[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         V[j],V[j-1]=V[j-1],V[j]

      if   EP[j]==EP[j-1]   and   ON[j]<ON[j-1]:
         EP[j],EP[j-1]=EP[j-1],EP[j]
         ON[j],ON[j-1]=ON[j-1],ON[j]
         V[j],V[j-1]=V[j-1],V[j]
```

Εμφανίζει πόσες πόλεις περιέχουν το γράμμα "Τ" ή το "t"

```
POL=["Athens", "Paris", "London", "Tirana", "Pristina"]
c=0
for x in POL:
   if 'T' in x or 't' in x:
      c=c+1
print c
```

Εμφανίζει πόσες πόλεις ξεκινούν με το κείμενο "ΝΕΑ"

```
c=0
POL=["ATHINA", "NEA AGXIALOS", "PATRA", "NEA IONIA", "LARISA"]
for i in range(len(POL)):
   ONOMA=POL[i]
   if ONOMA[ :3] == "NEA":
      c=c+1
```

Η **συνάρτηση binarySearch(lista, key)** η οποία επιστρέφει τη θέση του **key** μέσα στο **lista.**
Αν δεν υπάρχει το key επιστρέφει **-1**

```
def    binarysearch(EP, eponimo) :
   first = 0
   last = len(EP)-1
   found = False
   while  found== False  and  first <= last  :

      mid = ( first + last ) // 2
      if   EP[ mid ] == eponimo :
         found = True
      elif  EP[ mid ] < eponimo :
         first = mid + 1
      else:
         last = mid-1

   if  found == True:
       return  mid
  else:
       return  -1
```

Γεμίζουμε μια λίστα Α με τα γράμματα μιας συμβολοσειράς

```
A=[]

B=str(raw_input("dose leksi"))

A=list(B)
```

```
A=[]

B=str(raw_input("dose leksi"))
for  x in   B:
    A.append(x)
```

Δίνουμε από το πληκτρολόγιο το όνομα ενός προϊόντος και αν υπάρχει τότε μας τυπώνει τα τεμάχια του προϊόντος αλλιώς τυπώνει μήνυμα ότι δεν υπάρχει

```
ON=[“Domates”, “Patates”, “Piperies”, “Arakas”]
T= [ 100, 520, 86, 120 ]

Y=True
Onoma=raw_input('Dose onoma proiontos')
for x in range(4):
   if  ON[x]==Onoma:
      print T[x]
      Y=False

if Y==True:
   print "Den yparxei"
```

Γεμίζει και αδειάζει μια **ΣΤΟΙΒΑ** από το **ΤΕΛΟΣ**

```
A=[]
x=input(‘Dose timi’)
while   x !=0:
    A.append(x)
    x=input(‘Dose nea  timi’)

while  A!=[]:
    A.pop()
```

Γεμίζει και αδειάζει μια **ΣΤΟΙΒΑ** από την **ΑΡΧΗ**

```
A=[]
x=input(‘Dose timi’)
while   x !=0:
    A.insert(0, x)
    x=input(‘Dose nea  timi’)

while  A!=[]:
    A.pop(0)
```

Γεμίζει και **ΟΥΡΑ** από το **ΤΕΛΟΣ** και την αδειάζει από την **ΑΡΧΗ**

```
A=[]
x=input(‘Dose timi’)
while   x !=0:
    A.append(x)
    x=input(‘Dose nea  timi’)

while  A!=[]:
    A.pop(0)
```

```
A=[]
x=input(‘Dose timi’)
while   x !=0:
    A = A + [x]
    x=input(‘Dose nea  timi’)

while  A!=[]:
    A.pop()
```

```
A=[]
x=input(‘Dose timi’)
while   x !=0:
    A = [x] + A
    x=input(‘Dose nea  timi’)

while  A!=[]:
    A.pop(0)
```

```
A=[]
x=input(‘Dose timi’)
while   x !=0:
    A = A + [x]
    x=input(‘Dose nea  timi’)

while  A!=[]:
    A.pop(0)
```

```
x=str(raw_input("dose leksi"))
while   x!="TELOS" and x!="telos" :

        x=str(raw_input("dose leksi"))
```

```
x=str(raw_input("dose leksi"))
while   x not in ["TELOS" , “telos"] :

        x=str(raw_input("dose leksi"))
```

X=input('Dose vathmo') while **X<0 or X>20** : X=input('Dose ksana ton sosto vathmo metaksi 0 -20')	**Έλεγχος ορθότητας τιμών**

X=[10, 15, 20, 25] Y=['dog', 'cat', 'human'] Z=[X, Y] print **Z [1] [2]**	Η Z=[[10, 15, 20, 25], ['dog', 'cat', 'human']] Οπότε print Z [1] [2] το 1 σημαίνει την λίστα ['dog', 'cat', 'human'] και το 2 από την λίστα αυτή το στοιχείο στην θέση 2 δηλαδή **human**

A=[13, 15, 12, 18] A[0]=11	#λίστα #αντικαθιστά το 13 με το 11	B=A **C=A[:]**	# Η λίστα Β ταυτίζεται με την λίστα Α # Η λίστα C είναι **αντίγραφο** της Α

Ταξινόμηση Ευθείας ανταλλαγής στα στοιχεία μιας **συμβολοσειρά**	

B='unix LINUX'
A=[]
for x in B:
  A.append(x)
 	
N=len(A)
for i in range (1, N, 1):
  for j in range (N-1, i-1, -1):
  if A[j] < A[j-1] :
  A[j], A[j-1]=A[j-1],A[j]
S=''
for x in A:
  S=S+x
	B='unix LINUX'
	A=list(B)
	
	N=len(A)
	for i in range (1, N, 1):
	  for j in range (N-1, i-1, -1):
	  if A[j] < A[j-1] :
	  A[j], A[j-1]=A[j-1],A[j]
	S=''
	for x in A:
	  S=S+x

Υπολογισμός Ποσοστού %

Μια τάξη έχει 20 μαθητές και από αυτούς οι 10 έχουν λιγότερες από 40 απουσίες. Ποιο είναι το ποσοστό επί της 100 των μαθητών ;

Ποσοστό μαθητών με λιγότερες από 40 απουσίες = 100 * 10 / 25

20 → 10

100 → ;

Στους 20 είναι 10

Στους 100 πόσοι είναι ;

Έχουμε το ποσοστό και υπολογίζουμε τον αριθμό των μαθητών

Σε μία τάξη το 50 % των μαθητών έχει λιγότερες από 40 απουσίες. Η τάξη έχει 20 μαθητές. πόσοι έχουν λιγότερες από 40 απουσίες ;

Μαθητές με λιγότερες από 40 απουσίες = 20 *50 / 100

100 → 50

20 → ;

Στους 100 είναι 50

Στους 20 πόσοι είναι ;

Συνάρτηση η οποία θα δέχεται μια λίστα, θα ελέγχει αν τα στοιχεία της είναι σε αύξουσα σειρά και θα επιστρέφει αντίστοιχα True ή False.

```
def   Elenxos(A):
      F = True
      i = 0
      N = len(A)
      while F==True  and  i<N-1 :
            if (A[ i ] > A[ i+1 ]) :
                  F = False
            i = i +1
     return    F
```

Συνάρτηση η οποία θα δέχεται μια λίστα, θα ελέγχει αν τα στοιχεία της είναι σε αύξουσα σειρά ή φθίνουσα σειρά και θα επιστρέφει αντίστοιχα τον τύπο ταξινόμησης. Αν δεν είναι ταξινομημένη επιστρέφει κατάλληλο μήνυμα

```
def   Elenxos(A):
      F = True
      i = 0
      N = len(A)
      if A[0]< A[1]:

         while F==True  and  i<N-1 :
            if (A[ i ] > A[ i+1 ]) :
               F = False
            i = i +1
         if F==True:
            return "Afksousa"
         else:
            return    “Den einai taksinomimeni”
      if A[0]> A[1]:

         while F==True  and  i<N-1 :
            if (A[ i ] < A[ i+1 ]) :
               F = False
            i = i +1
         if F==True:
            return "fthinousa"
         else:
            return    “Den einai taksinomimeni”

A=[15, 18, 19, 22, 28, 42, 50]

B=[50, 42, 28, 22, 19, 18, 15]

print Elenxos(A)
print Elenxos(B)
```