



# Ελπύνωρ

ΠΙΛΟΤΙΚΗ ΑΞΙΟΠΟΙΗΣΗ ΔΙΚΤΥΑΚΗΣ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΥΠΟΔΟΜΗΣ ΣΕ ΕΝΙΑΙΑ ΛΥΚΕΙΑ

## Τετράδιο Μαθητή για το λογισμικό Περιβάλλον Δομημένου Προγραμματισμού



Ελληνικά  
Σχολεία  
στην  
Κοινωνία  
της  
Πληροφορίας

Φορέας  
Υλοποίησης

Ερευνητικό  
Ακαδημαϊκό  
Ινστιτούτο  
Τεχνολογίας  
Υπολογιστών

Ανάδοχοι

conceptum A.E.

Φορείς της Ενέργειας

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΔΙΕΥΘΥΝΣΗ ΣΠΟΥΔΩΝ ΔΕΥΤΕΡΟΒΑΘΜΙΑΣ ΕΚΠΑΙΔΕΥΣΗΣ  
ΔΙΕΥΘΥΝΣΗ ΚΟΙΝΟΤΙΚΟΥ ΠΛΑΙΣΙΟΥ ΣΤΗΡΙΞΗΣ



Παιδαγωγικό  
Ινστιτούτο

Συγχρηματοδότηση από την Ευρωπαϊκή Ένωση



**Συντελεστές**

Υπεύθυνη έργου ΕΛΠΗΝΩΡ:

Μαρία Σκιαδέλλη – EAITY

Σύνδεσμος Παιδαγωγικού Ινστιτούτου:

Αναστασία Χαρχαρίδου

Υπεύθυνος υποέργου από τη μεριά του Αναδόχου

και επιστημονικός υπεύθυνος:

Αδάμ Κ. Δαμιανάκης

Συγγραφή:

Αναστασία Γ. Αξαρίδου

Φιλολογική επιμέλεια:

Ιωάννα Μόσχου

Εικόνες:

Εμμανουήλ Σφενδύλης

Επικοινωνία:

 **conceptum A.E.**

Πατησίων 95

Αθήνα 10434

τηλ: 010 88 38 858, φαξ: 010 88 38 691

[www.conceptum.gr](http://www.conceptum.gr), email: [support@conceptum.gr](mailto:support@conceptum.gr)

## Περιεχόμενα

1. Εισαγωγή .....	6
2. Δραστηριότητες .....	7
Δραστηριότητα 1: Εφαρμογή βασικών αλγοριθμικών εννοιών για τη δημιουργία απλών προγραμμάτων .....	7
Δραστηριότητα 1α: Εκχώρηση τιμών σε μεταβλητές και εμφάνιση του αποτελέσματος.....	7
Δραστηριότητα 1β: Δομές επανάληψης .....	8
Δραστηριότητα 1γ: Πίνακες – Εισαγωγή και ανάκτηση τιμών των στοιχείων .....	10
Δραστηριότητα 1δ: Εμφωλευμένες εντολές – Δομές επιλογής – Είσοδος δεδομένων .....	11
Δραστηριότητα 2: Επίλυση του τριωνύμου $Ax^2+Bx+C = 0$ .....	14
Δραστηριότητα 3: Δημιουργία σχήματος «δέντρου» που αποτελείται από X αριθμό γραμμών, με χρήση ενός χαρακτήρα (κάθετη γραμμή « »). .....	16
Δραστηριότητα 4: Ταξινόμηση των στοιχείων μονοδιάστατου πίνακα N στοιχείων κατά αύξουσα σειρά με τη μέθοδο «ταξινόμηση φουσαλίδας» (bubble sort).....	18



# 1. Εισαγωγή

Στο Τετράδιο αυτό περιέχονται 4 κύριες δραστηριότητες που μπορούν να εφαρμοστούν στο Περιβάλλον Δομημένου Προγραμματισμού (ΠΔΠ), με απώτερο στόχο να κατανοήσετε τη λογική ανάλυση και τη δημιουργία του αλγόριθμου ενός προβλήματος, να εξοικειωθείτε με την υλοποίηση και την εκτέλεση προγραμμάτων στη μορφή εντολών ψευδοκώδικα και τέλος με την εξαγωγή αποτελεσμάτων. Κατά τη δημιουργία των προγραμμάτων μπορείτε να παρακολουθείτε ταυτόχρονα στο ειδικό παράθυρο το λογικό διάγραμμα ροής του προγράμματος, καθώς επίσης και τη μορφή που έχει το πρόγραμμα σε κώδικα Basic.

Κάθε δραστηριότητα αποτελείται από τις ακόλουθες ενότητες:

- διδακτικοί στόχοι
- ζητούμενο
- βοηθητική υπόδειξη (όπου απαιτείται)
- λογική ανάλυση ή μαθηματική λύση (όπου απαιτείται)
- αλγόριθμος

Για το πώς μπορείτε να εργαστείτε στο ΠΔΠ, συμβουλευτείτε το εγχειρίδιο χρήσης.

## 2. Δραστηριότητες

### **Δραστηριότητα 1: Εφαρμογή βασικών αλγοριθμικών εννοιών για τη δημιουργία απλών προγραμμάτων**

Η δραστηριότητα αυτή αποτελείται από 4 επιμέρους δραστηριότητες, οι οποίες θα σας βοηθήσουν να κατανοήσετε την έννοια της λογικής ακολουθίας στη δημιουργία ενός προγράμματος με την τοποθέτηση των προτάσεων-εντολών σε λογική σειρά εκτέλεσης, καθώς επίσης και να εξοικειωθείτε σταδιακά με τη χρήση θεμελιωδών δομών και εντολών του προγραμματισμού.

Πιο αναλυτικά, στη δραστηριότητα 1α καλείστε να κάνετε εκχώρηση τιμών σε μεταβλητές και στο τέλος να εμφανίσετε τα αποτελέσματα· στην 1β γίνεται επιπλέον η χρήση μιας εντολής επανάληψης και αριθμητικών πράξεων· στην 1γ, εκτός από τα παραπάνω, εισάγεται η χρήση της δομής ενός πίνακα και η διαχείριση των στοιχείων του· και, τέλος, στην 1δ γίνεται η χρήση εμφωλευμένων εντολών και συγκεκριμένα η δομή μιας εντολής απόφασης μέσα σε μια δομή επανάληψης.

#### **Δραστηριότητα 1α: Εκχώρηση τιμών σε μεταβλητές και εμφάνιση του αποτελέσματος**

##### **Διδακτικοί στόχοι**

Στόχος της δραστηριότητας αυτής είναι να μάθετε:

- Να χρησιμοποιείτε τις μεταβλητές σε ένα πρόγραμμα
- Να εκχωρείτε τιμή σε μεταβλητή
- Να χρησιμοποιείτε τον τύπο της συμβολοσειράς
- Να εμφανίζετε τα αποτελέσματα ενός προγράμματος

##### **Ζητούμενο**

Να κατασκευαστεί ο ψευδοκώδικας του προγράμματος το οποίο εκχωρεί στις μεταβλητές ΟΝΟΜΑ, ΕΡΩΝΥΜΟ, ΗΛΙΚΙΑ, ΤΑΚΣΗ, ΤΜΗΜΑ τα προσωπικά σας στοιχεία και εμφανίζει τα αποτελέσματα το ένα κάτω από το άλλο στο παράθυρο των αποτελεσμάτων. Τέλος, αποθηκεύστε το πρόγραμμα με το όνομα **Ask\_1a.elp**, εκτελέστε το και δείτε τα αποτελέσματα.

##### **Υπόδειξη**

I. Για να κάνετε απλά εκχώρηση τιμής σε μια μεταβλητή, χρησιμοποιήστε την εντολή:

-Θέσε .. ίσο με ..

Στις μεταβλητές της δραστηριότητας θα γίνει εκχώρηση τιμής τύπου αλφαριθμητικής παράστασης (συμβολοσειρά). Για να ενημερώσουμε το πρόγραμμα ότι θα εισάγουμε μια τιμή με αυτό τον τύπο, αρκεί να προσθέσουμε στην αρχή και στο τέλος της τιμής το χαρακτήρα των απλών εισαγωγικών (") (quote), π.χ. "συμβολοσειρά".

II. Για να εμφανίσετε τις μεταβλητές τη μια κάτω από την άλλη, χρησιμοποιήστε την εντολή:

-Εμφάνισε (..) και άλλαξε γραμμή

##### **Αλγόριθμος**

Ένας αλγόριθμος που περιγράφει τη διαδικασία εμφάνισης των προσωπικών σας στοιχείων είναι:

- Βάλε στις μεταβλητές ONOMA, EPWNYMO, HLIKIA, TAKSH, TMHMA τις αντίστοιχες τιμές.
- Εμφάνισε τα περιεχόμενα των μεταβλητών ONOMA, EPWNYMO, HLIKIA, TAKSH, TMHMA, το ένα κάτω από το άλλο.

### Δραστηριότητα 1β: Δομές επανάληψης

#### Διδακτικοί στόχοι

Στόχος της δραστηριότητας αυτής είναι να μάθετε:

- Να χρησιμοποιείτε τη δομή επανάληψης
- Να χρησιμοποιείτε τον τύπο δεδομένων των ακέραιων αριθμών
- Να εφαρμόζετε τις αριθμητικές πράξεις

#### Ζητούμενο Α΄

Να κατασκευαστεί ο ψευδοκώδικας του προγράμματος που υπολογίζει το τετράγωνο όλων των ακέραιων αριθμών από το 1 έως το 20 και εμφανίζει τα αποτελέσματα, το ένα κάτω από το άλλο. Αποθηκεύστε το πρόγραμμα με το όνομα **Ask\_1b\_1.elp**, εκτελέστε το και δείτε τα αποτελέσματα.

#### Υπόδειξη

- I. Για να επαναλάβετε την ίδια διαδοχή εντολών πολλές φορές μέσα σε ένα πρόγραμμα, χρησιμοποιήστε τη δομή:

-Για .. από .. μέχρι .. κάνε

(εντολή)

(εντολή)

.....

(εντολή)

ΤέλοςΓια

Στο πρώτο κενό της εντολής αυτής εισάγεται μια μεταβλητή που λειτουργεί σαν μετρητής των επαναλήψεων. Η μεταβλητή αυτή μπορεί να πάρει μόνο ακέραιες τιμές μεταξύ δύο οριακών ακέραιων τιμών, αρχική και τελική, οι οποίες εισάγονται αντίστοιχα στο δεύτερο και τρίτο κενό της εντολής. Δηλαδή, στα δύο τελευταία κενά μπορούμε να εισάγουμε είτε ακέραιους αριθμούς είτε μεταβλητές στις οποίες προηγουμένως έχουν εκχωρηθεί ακέραιες τιμές. Στη συγκεκριμένη εντολή επανάληψης ο μετρητής αυξάνεται, και γι' αυτό η αρχική τιμή πρέπει να είναι μικρότερη ή ίση της τελικής. Στην περίπτωση της ισότητας η επαναληπτική διαδικασία θα εκτελεστεί μόνο μία φορά. Όταν ο μετρητής γίνει ίσος με την τελική τιμή (ή για κάποιο λόγο κατά την εκτέλεση του προγράμματος γίνει μεγαλύτερος από αυτήν), τότε η επαναληπτική διαδικασία τερματίζεται και το πρόγραμμα εκτελεί την επόμενη εντολή.

- II. Στον προγραμματισμό οι αριθμητικές πράξεις επιτρέπονται μόνο μεταξύ ακέραιων ή πραγματικών αριθμών ή μεταβλητών αυτού του τύπου. Για να κάνουμε μια αριθμητική πράξη μεταξύ δύο αριθμητικών μεταβλητών ή αριθμών, μπορούμε είτε να δημιουργήσουμε μια αριθμητική παράσταση με τη χρήση των αριθμητικών τελεστών, π.χ.

-Θέσε X ίσο με  $a+3$



ή να χρησιμοποιήσουμε μια από τις εντολές εκχώρησης τιμής που περιλαμβάνει έναν αριθμητικό τελεστή, π.χ.

-Θέσε  $X$  ίσο με το άθροισμα του  $a$  και του  $3$

III. Τέλος, για να εμφανίσετε τα αποτελέσματα το ένα κάτω από το άλλο χρησιμοποιήστε την εντολή:

-Εμφάνισε ( .. ) και άλλαξε γραμμή

### **Αλγόριθμος**

Ο αλγόριθμος που περιγράφει διαδικασία υπολογισμού του ζητούμενου είναι:

- Ξεκινώντας από το 1, επανάλαβε 20 φορές τα παρακάτω βήματα:
  - Βάλε στη μεταβλητή TETRAGWNO το αποτέλεσμα της πράξης  $J * J$ , όπου  $J$  ο μετρητής της επαναληπτικής διαδικασίας.
  - Εμφάνισε σε νέα γραμμή το περιεχόμενο της μεταβλητής TETRAGWNO.

### **Ζητούμενο Β'**

Να κατασκευαστεί ο ψευδοκώδικας του προγράμματος που υπολογίζει το τετράγωνο όλων των μονών ακέραιων αριθμών από το 1 μέχρι το 20 και εμφανίζει τα αποτελέσματα, το ένα κάτω από το άλλο. Αποθηκεύστε το πρόγραμμα με το όνομα **Ask\_1b\_2.elp**, εκτελέστε το και δείτε τα αποτελέσματα.

### **Υπόδειξη**

Βλ. Υπόδειξη του Ζητούμενου Α'.

### **Αλγόριθμος**

Ο αλγόριθμος που περιγράφει διαδικασία υπολογισμού του ζητούμενου είναι:

- Ξεκινώντας από το 1, επανάλαβε 20 φορές τα παρακάτω βήματα:
  - Βάλε στη μεταβλητή TETRAGWNO το αποτέλεσμα της πράξης  $J * J$ , όπου  $J$  ο μετρητής της επαναληπτικής διαδικασίας.
  - Εμφάνισε σε νέα γραμμή το περιεχόμενο της μεταβλητής TETRAGWNO.
  - Αύξησε το μετρητή  $J$  κατά 1.

### **Ζητούμενο Γ'**

Να κατασκευαστεί ο ψευδοκώδικας του προγράμματος που υπολογίζει το τετράγωνο όλων των ζυγών ακέραιων αριθμών από το 1 μέχρι το 20 και εμφανίζει τα αποτελέσματα, το ένα κάτω από το άλλο. Αποθηκεύστε το πρόγραμμα με το όνομα **Ask\_1b\_3.elp**, εκτελέστε το και δείτε τα αποτελέσματα.

### **Υπόδειξη**

Βλ. Υπόδειξη του Ζητούμενου Α'.

### **Αλγόριθμος**

Ο αλγόριθμος που περιγράφει διαδικασία υπολογισμού του ζητούμενου είναι:

- Ξεκινώντας από το 1, επανάλαβε 20 φορές τα παρακάτω βήματα:

- Αύξησε το J κατά 1, όπου J ο μετρητής της επαναληπτικής διαδικασίας.
- Βάλε στη μεταβλητή TETRAGWNO το αποτέλεσμα της πράξης J\*J.
- Εμφάνισε σε νέα γραμμή το περιεχόμενο της μεταβλητής TETRAGWNO.

### Δραστηριότητα 1γ: Πίνακες – Εισαγωγή και ανάκτηση τιμών των στοιχείων

#### Διδακτικοί στόχοι

Στόχος της δραστηριότητας αυτής είναι να μάθετε:

- Να χρησιμοποιείτε έναν πίνακα
- Να διαχειρίζεστε (εκχώρηση-ανάκτηση τιμής) τα στοιχεία ενός πίνακα
- Να σαρώσετε σειριακά τα στοιχεία ενός πίνακα

#### Ζητούμενο

Να κατασκευαστεί ο ψευδοκώδικας του προγράμματος που δημιουργεί έναν κενό πίνακα, με το όνομα PINAKAS, 20 στοιχείων, και στη συνέχεια αναθέτει στα στοιχεία του πίνακα αυτού τα τετράγωνα των ακέραιων αριθμών από 1 μέχρι 20. Τέλος, να γίνεται η εμφάνιση των τιμών όλων των στοιχείων του πίνακα, η μια κάτω από την άλλη. Αποθηκεύστε το πρόγραμμα με το όνομα Ask\_1c.elp, εκτελέστε το και δείτε τα αποτελέσματα.

#### Υπόδειξη

I. Για να δημιουργήσετε έναν κενό πίνακα χρησιμοποιήστε την εντολή:

*-Δήλωση πίνακα .. με αριθμό στοιχείων ..*

Στο πρώτο κενό της παραπάνω εντολής εισάγεται το όνομα της μεταβλητής με το οποίο γίνεται αναφορά στον πίνακα. Στο δεύτερο κενό εισάγεται ένας ακέραιος (ή μεταβλητή αυτού του τύπου) που δηλώνει το μέγιστο πλήθος των στοιχείων του πίνακα.

II. Για να αναφερθούμε στο x-οστό στοιχείο ενός πίνακα εισάγουμε ONOMA\_ΠΙΝΑΚΑ[x]. Την αναφορά αυτή τη χρησιμοποιούμε είτε για να εκχωρήσουμε μια τιμή στο συγκεκριμένο στοιχείο του πίνακα, είτε για να ανακτήσουμε την τιμή του.

Π.χ. Έστω ο πίνακας PIN ο οποίος περιέχει 5 στοιχεία.

Μπορούμε να αναθέσουμε την τιμή 45 στο δεύτερο στοιχείο του ως εξής:

*-Θέσε PIN[2] ίσο με 45,*

είτε να ανακτήσουμε (με τη βοήθεια μιας μεταβλητής) την τιμή του τρίτου στοιχείου του ως εξής:

*-Θέσε METABLHTH ίσο με PIN[2]*

III. Για να σαρώσουμε σειριακά όλα τα στοιχεία ενός πίνακα χρησιμοποιούμε μια επαναληπτική δομή, στο εσωτερικό της οποίας γίνεται αναφορά στα στοιχεία του πίνακα, με τη βοήθεια του μετρητή της.

#### Αλγόριθμος

Ο αλγόριθμος που περιγράφει τη ζητούμενη διαδικασία είναι:

- Δημιούργησε έναν κενό πίνακα, με το όνομα PINAKAS, 20 στοιχείων.
- Ξεκινώντας από το 1, επανάλαβε 20 φορές τα παρακάτω βήματα:
  - Βάλε στη μεταβλητή TETRAGWNO το αποτέλεσμα της πράξης J\*J, όπου J ο μετρητής της επαναληπτικής διαδικασίας.

- Βάλε στο J-οστό στοιχείο του πίνακα PINAKAS το περιεχόμενο της μεταβλητής TETRAGWNO.
- Ξεκινώντας από το 1, επανάλαβε 20 φορές τα παρακάτω βήματα:
  - Εμφάνισε σε νέα γραμμή το K-οστό στοιχείο του πίνακα PINAKAS, όπου K ο μετρητής της επαναληπτικής διαδικασίας.

### Δραστηριότητα 1δ: Εμφωλευμένες εντολές – Δομές επιλογής – Είσοδος δεδομένων

### *Διδακτικοί στόχοι*

Στόχος της δραστηριότητας αυτής είναι να μάθετε:

- Να χρησιμοποιείτε τις εμφωλευμένες δομές
- Να χρησιμοποιείτε τις λογικές συνθήκες και τις δομές επιλογής (απόφασης)
- Να εμφανίζετε τα δεδομένα, διαχωρισμένα με κάποιο χαρακτήρα, στην ίδια γραμμή
- Να εισάγετε δεδομένα σε ένα πρόγραμμα κατά την εκτέλεσή του

**Ζητούμενο Α΄**

Να κατασκευαστεί ο ψευδοκώδικας του προγράμματος το οποίο βρίσκει και εμφανίζει σε μια γραμμή (διαχωρίζοντας με «.») όλους τους ακέραιους αριθμούς μεταξύ 20 και 55 που είναι πολλαπλάσια του τρία. Στη συνέχεια, αποθηκεύστε το πρόγραμμα ως **Ask\_1d\_1.elp** και εκτελέστε το για να δείτε τα αποτελέσματα.

## Υπόδειξη

- I. Κάθε λογική συνθήκη επιστρέφει (έχει ως αποτέλεσμα) δύο δυνατές τιμές: Αληθής (True ή 1) ή Ψευδής (False ή 0). Μια λογική συνθήκη μπορεί να είναι είτε μια παράσταση στην οποία γίνεται σύγκριση μεταξύ δύο τιμών ή μεταβλητών με τη βοήθεια ενός συγκριτικού τελεστή, ή μια μεταβλητή λογικού τύπου (Boolean). Οι συγκριτικοί τελεστές είναι:

Ισότητας (=), ανισότητας ( $\neq$ ), μικρότερου (<), μικρότερου ή ίσου ( $\leq$ ), μεγαλύτερου (>), μεγαλύτερου ή ίσου ( $\geq$ ).

Για να δημιουργήσουμε μια λογική συνθήκη σχηματίζουμε μια παράσταση που έχει δύο μέλη εκατέρωθεν ενός συγκριτικού τελεστή. Π.χ.

Var1 <> Var2	(αληθής όταν τα Var1 και Var2 δεν έχουν την ίδια τιμή)
--------------	---

“internet” <= “my internet” (ψευδής)

“internet” <= “m” (ψευδής)

$$2+1-2*1 < 3 \quad (\alpha\lambda\eta\theta\acute{\eta}\varsigma)$$

- II. Για να εμφανίσουμε τα αποτελέσματα σε μια γραμμή και όχι το ένα κάτω από το άλλο, χρησιμοποιούμε την εντολή:

-Εμφάνισε ( .. )

Το αποτέλεσμα της πολλαπλής χρήσης της εντολής αυτής είναι μια συνεχόμενη ακολουθία χαρακτήρων στην οποία δε διακρίνονται τα αποτελέσματα.

Π.χ. Το αποτέλεσμα της εκτέλεσης του ψευδοκώδικα:

-Εμφάνισε ( "324" )

-Εμφάνισε ( "56" )

-Εμφάνισε ( "67" )

-Εμφάνισε ( "105" )

είναι: «3245667105»

Στην περίπτωση που θέλουμε να διαχωρίσουμε τα αποτελέσματα, φροντίζουμε να εμφανιστεί μαζί με αυτά και κάποιος διαχωριστικός χαρακτήρας (ή συμβολοσειρά), π.χ. το κενό « », έτσι ώστε τα αποτελέσματα να είναι στην ίδια γραμμή μεν, ευδιάκριτα δε. Έτσι, στο παραπάνω παράδειγμα, αν έχουμε:

-Εμφάνισε ( "324", " " )

-Εμφάνισε ( "56", " " )

-Εμφάνισε ( "67" )

-Εμφάνισε ( " " )

-Εμφάνισε ( "105" )

Το αποτέλεσμα της εκτέλεσης θα είναι: «324 56 67 105».

### Αλγόριθμος

Ο αλγόριθμος που περιγράφει τη ζητούμενη διαδικασία είναι:

- Βάλε στη μεταβλητή K την τιμή μηδέν (αρχικοποίηση μεταβλητής).
- Δημιούργησε έναν κενό πίνακα, με το όνομα PINAKAS, 35 (= 55-20) στοιχείων.
- Ξεκινώντας από το 20, επανάλαβε 35 φορές τα παρακάτω βήματα:
  - Βάλε στη μεταβλητή ΥΠΟΛΟΙΠΟ το υπόλοιπο της διαίρεσης J:3, όπου J ο μετρητής της επαναληπτικής διαδικασίας.
  - Εξέτασε το περιεχόμενο της μεταβλητής ΥΠΟΛΟΙΠΟ.
    - αν το ΥΠΟΛΟΙΠΟ είναι ίσο με μηδέν (δηλαδή αν η διαίρεση του προηγούμενου βήματος ήταν τέλεια), τότε:
      - i. Αύξησε τη μεταβλητή K κατά 1.
      - ii. Βάλε στο K-οστό στοιχείο του πίνακα PINAKAS το περιεχόμενο της μεταβλητής J.
- Ξεκινώντας από το 1, επανάλαβε K φορές τα παρακάτω βήματα:
  - Εμφάνισε το F-οστό στοιχείο του πίνακα PINAKAS, όπου F ο μετρητής της επαναληπτικής διαδικασίας
  - Εξέτασε τη συνθήκη  $F < K$ 
    - αν η συνθήκη είναι Αληθής τότε:
      - i. Εμφάνισε τους χαρακτήρες κόμμα+κενό (διαχωριστική συμβολοσειρά).

**Ζητούμενο Β΄**

Να τροποποιηθεί ο ψευδοκώδικας του Ζητουμένου Α΄, έτσι ώστε το πρόγραμμα να γίνει πιο δυναμικό: να γίνεται η εύρεση των πολλαπλάσιων ενός τυχαίου αριθμού X, τα οποία είναι μεταξύ δύο οποιωνδήποτε ακέραιων αριθμών A και B (με  $A > B$ ). Τέλος, αποθηκεύστε το τροποποιημένο πρόγραμμα ως **Ask\_1d\_2.eip**. Εκτελέστε το και δείτε τα αποτελέσματα.

**Υπόδειξη**

- I. Βλ. Υπόδειξη Ζητουμένου Α΄.
- II. Για να εισάγουμε τιμές στις μεταβλητές ενός προγράμματος κατά τη διάρκεια της εκτέλεσης, χρησιμοποιούμε την εντολή:

-Διάβασε ..

Στο κενό της εντολής αυτής εισάγουμε το όνομα της μεταβλητής στην οποία θέλουμε να εκχωρήσουμε τιμή. Μπορούμε επίσης στην ίδια εντολή να εισάγουμε περισσότερες από μία μεταβλητές, διαχωρισμένες μεταξύ τους με κόμματα, και στη συνέχεια να εισάγουμε με την ίδια σειρά τις τιμές τους στο παράθυρο εισαγωγής τιμών που θα εμφανίσει το πρόγραμμα την ώρα της εκτέλεσης.

**Αλγόριθμος**

Ο αλγόριθμος που περιγράφει τη ζητούμενη διαδικασία είναι:

- Διάβασε τις μεταβλητές A, B, X.
- Βάλε στη μεταβλητή K το μηδέν (αρχικοποίηση μεταβλητής).
- Δημιούργησε έναν κενό πίνακα, με το όνομα PINAKAS, B-A στοιχείων.
- Ξεκινώντας από το A, επανάλαβε B-A φορές τα παρακάτω βήματα:
  - Βάλε στη μεταβλητή ΥΠΟΛΟΙΠΟ το υπόλοιπο της διαίρεσης J:X, όπου J ο μετρητής της επαναληπτικής διαδικασίας.
  - Εξέτασε το περιεχόμενο της μεταβλητής ΥΠΟΛΟΙΠΟ.
    - αν το ΥΠΟΛΟΙΠΟ είναι ίσο με μηδέν (δηλαδή αν η διαίρεση του προηγούμενου βήματος ήταν τέλεια), τότε:
      - i. Αύξησε τη μεταβλητή K κατά 1.
      - ii. Βάλε στο K-οστό στοιχείο του πίνακα PINAKAS το περιεχόμενο της μεταβλητής J.
- Ξεκινώντας από το 1, επανάλαβε K φορές τα παρακάτω βήματα:
  - Εμφάνισε το F-οστό στοιχείο του πίνακα PINAKAS, όπου F ο μετρητής της επαναληπτικής διαδικασίας.
  - Εξέτασε τη συνθήκη  $F < K$ 
    - αν η συνθήκη είναι Αληθής τότε:
      - i. εμφάνισε τους χαρακτήρες κόμμα+κενό (διαχωριστική συμβολοσειρά).

## Δραστηριότητα 2: Επίλυση του τριωνύμου $Ax^2+Bx+C=0$

### Διδακτικοί στόχοι

Στόχος της δραστηριότητας αυτής είναι να μάθετε:

- Να αναλύετε ένα σύνθετο πρόβλημα σε λογικά βήματα.
- Να δημιουργείτε τον αλγόριθμο για την επίλυση ενός μαθηματικού προβλήματος.
- Να χρησιμοποιείτε τις εντολές εκχώρησης (αποτελέσματος αριθμητικών πράξεων και μαθηματικών συναρτήσεων) σε μεταβλητές, καθώς την εντολής απόφασης.

### Ζητούμενο

Να δημιουργηθεί ο αλγόριθμος και στη συνέχεια ο ψευδοκώδικας του προγράμματος που υπολογίζει τις ρίζες του τριωνύμου  $Ax^2+Bx+C=0$ . Αποθηκεύστε το πρόγραμμα με το όνομα **Ask\_2.elp** και εκτελέστε το για να δείτε τα αποτελέσματα.

### Υπόδειξη

- Για να μπορούμε με το ίδιο πρόγραμμα να υπολογίζουμε τις ρίζες τυχαίου τριωνύμου, χρησιμοποιούμε την εντολή *Διάβασε* στην αρχή του προγράμματος, ώστε να εισάγουμε τις τιμές των συντελεστών του τριωνύμου που επιθυμούμε κάθε φορά. Το πρόγραμμα γίνεται με αυτό τον τρόπο δυναμικό και μπορεί να καλύψει τη γενικότερη περίπτωση ενός προβλήματος. Διαφορετικά, θα πρέπει να δημιουργήσουμε άλλο πρόγραμμα για κάθε νέο τριώνυμο του οποίου θέλουμε να υπολογίσουμε τις ρίζες.
- Για να μας επιστρέψει το πρόγραμμα αριθμητικά αποτελέσματα που να είναι πραγματικοί αριθμοί, θα πρέπει να προσέξουμε να εισάγονται στις εντολές του ψευδοκώδικα ορίσματα πραγματικών αριθμών. Έτσι για τον υπολογισμό των πραγματικών ριζών ενός τριωνύμου, θα πρέπει το όρισμα της εντολής *Διάβασε*, που είναι οι συντελεστές του τριωνύμου, να είναι ακολουθία πραγματικών αριθμών. Για να δηλώσουμε στο πρόγραμμα έναν αριθμό ως πραγματικό, αυτός θα πρέπει να είναι του τύπου κινητής υποδιαστολής, διαφορετικά θεωρείται ακέραιος. Π.χ. 4 ακέραιος και 4.0 πραγματικός.

### Μαθηματική λύση

Η μαθηματική λύση του προβλήματος αναλύεται στα ακόλουθα βήματα:

(α) Υπολογισμός της διακρίνουσας:  $D = B^2 - 4AC$

(β) Διάκριση περιπτώσεων ανάλογα με την τιμή της διακρίνουσας που υπολογίστηκε στο (α). Διακρίνουμε τις παρακάτω περιπτώσεις:

Αν  $D < 0$ , τότε δεν υπάρχουν πραγματικές ρίζες.

Αν  $D = 0$ , τότε υπάρχει μια διπλή ρίζα που δίνεται από τον τύπο:

$$R = \frac{-B}{2A}$$

Αν  $D > 0$ , τότε οι ρίζες του τριωνύμου δίνονται από τον τύπο:

$$R_{1,2} = \frac{-B \pm (D)^{1/2}}{2A}$$

### Αλγόριθμος

Ο αλγόριθμος που περιγράφει την παραπάνω μαθηματική λύση είναι:

- Διάβασε και βάλε στις μεταβλητές A, B, C τους αντίστοιχους συντελεστές της εξίσωσης.
- Βάλε στη μεταβλητή DIAKRINOYSA το αποτέλεσμα της πράξης  $(B*B)-(4*A*C)$ .
- Εξέτασε τη μεταβλητή DIAKRINOYSA.
  - Αν η μεταβλητή DIAKRINOYSA είναι αρνητική, τότε εμφάνισε «Δεν υπάρχουν πραγματικές ρίζες».
  - Αν η μεταβλητή DIAKRINOYSA είναι μηδέν, τότε:
    - i. Βάλε στη μεταβλητή RIZA το αποτέλεσμα της πράξης  $(-B/(2*A))$ ,
    - ii. και εμφάνισε «Υπάρχει μια διπλή ρίζα: » και το περιεχόμενο της μεταβλητής RIZA.
  - Αν η μεταβλητή DIAKRINOYSA είναι θετική, τότε:
    - i. Βάλε στη μεταβλητή DIAKRINOYSA\_RIZA την τετραγωνική ρίζα της μεταβλητής DIAKRINOYSA.
    - ii. Βάλε στη μεταβλητή RIZA1 το αποτέλεσμα της πράξης  $(-B + DIAKRINOYSA\_RIZA) / (2*A)$ .
    - iii. Βάλε στη μεταβλητή RIZA2 το αποτέλεσμα της πράξης  $(-B - DIAKRINOYSA\_RIZA) / (2*A)$ ,
    - iv. και εμφάνισε «Οι πραγματικές ρίζες είναι: » και το περιεχόμενο των μεταβλητών RIZA1 και RIZA2.

### Δραστηριότητα 3: Δημιουργία σχήματος «δέντρου» που αποτελείται από $X$ αριθμό γραμμών, με χρήση ενός χαρακτήρα (κάθετη γραμμή «|»).

#### Διδακτικοί στόχοι

Στόχος της δραστηριότητας αυτής είναι να μάθετε:

- Να δημιουργείτε τον αλγόριθμο για το σχεδιασμό ενός παραμετροποιημένου γραφικού σχήματος.
- Να χρησιμοποιείτε τις εντολές εκχώρησης αποτελέσματος αριθμητικών πράξεων σε μεταβλητές, καθώς και τις εμφωλευμένες εντολές επανάληψης.

#### Ζητούμενο

Να δημιουργηθεί ο αλγόριθμος και στη συνέχεια ο ψευδοκώδικας του προγράμματος που σχηματίζει ένα δέντρο  $X$  γραμμών από το χαρακτήρα pipeline «|». Αποθηκεύστε το πρόγραμμα με το όνομα `Ask_3.elp`. Εκτελέστε το και δείτε τα αποτελέσματα.

#### Υπόδειξη

Όπως είδαμε στις προηγούμενες δραστηριότητες, για να δημιουργηθεί μια νέα γραμμή στο παράθυρο των Αποτελεσμάτων, χρησιμοποιούμε την εντολή:

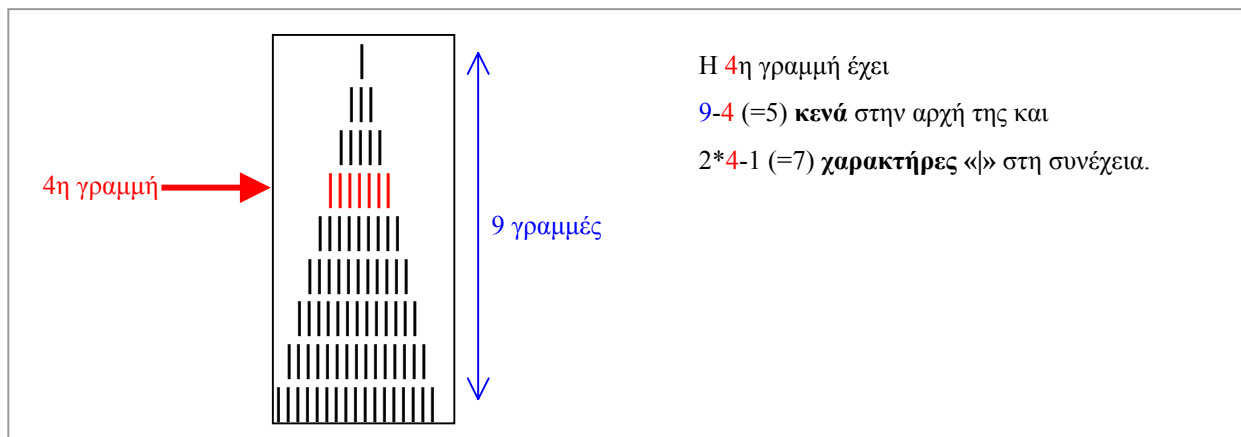
-Εμφάνισε ( .. ) και άλλαξε γραμμή

Κάθε φορά που γίνεται αλλαγή γραμμής, αυτό που συμβαίνει στην πραγματικότητα είναι η εισαγωγή στο τέλος της γραμμής ενός συγκεκριμένου κρυφού χαρακτήρα, του χαρακτήρα «αλλαγής γραμμής» (carriage return).

Με την παραπάνω εντολή το πρόγραμμα γράφει στην τρέχουσα γραμμή τα δεδομένα που επιστρέφει το όρισμα της εντολής (δηλαδή ό,τι έχει εισαχθεί στο κενό της) και στη συνέχεια προσθέτει το χαρακτήρα αλλαγής γραμμής. Την επόμενη φορά που θα εκτελεστεί κάποια εντολή εμφάνισης αποτελεσμάτων το πρόγραμμα θα αρχίσει να γράφει από εκεί που είχε σταματήσει, δηλαδή στην αρχή της νέας αυτής γραμμής που δημιουργήθηκε προηγουμένως, η οποία θεωρείται πλέον ως η τρέχουσα γραμμή. Για να κάνουμε μόνο αλλαγή γραμμής χωρίς να εξάγουμε κατ' ανάγκη αποτελέσματα, αρκεί να εισάγουμε ως όρισμα της εντολής την κενή συμβολοσειρά "":

-Εμφάνισε ( "" ) και άλλαξε γραμμή

#### Ανάλυση του προβλήματος



Αν είναι  $x$  ο αριθμός των γραμμών του δέντρου, για να σχηματιστεί το δέντρο γραμμή γραμμή, πρέπει σε κάθε γραμμή να γράφονται  $(x-i)$  συνεχόμενα κενά, όπου  $i$  ο αριθμός της τρέχουσας



γραμμής και στη συνέχεια στην ίδια γραμμή να γράφονται  $(2*i-1)$  συνεχόμενοι χαρακτήρες «|». Ακολουθεί ο χαρακτήρας αλλαγής γραμμής (carriage return). Η εγγραφή συνεχόμενων και όμοιων χαρακτήρων είναι επαναληπτική διαδικασία που υλοποιείται με εντολές επανάληψης.

### Αλγόριθμος

- Διάβασε και βάλε στη μεταβλητή  $X$  το πλήθος των γραμμών που θα σχηματίσουν το δέντρο.
- Ξεκινώντας από το 1, επανάλαβε  $X$  φορές τα παρακάτω βήματα:
  - Βάλε στη μεταβλητή  $KENA$  το αποτέλεσμα της πράξης  $X-1$ .
  - Βάλε στη μεταβλητή  $CHARACTERS$  το αποτέλεσμα της πράξης  $2*I-1$ , όπου  $I$  ο μετρητής της επαναληπτικής διαδικασίας.
  - Ξεκινώντας από το 1, επανάλαβε τόσες φορές όσες το περιεχόμενο της μεταβλητής  $KENA$  τα παρακάτω βήματα:
    - Εμφάνισε το χαρακτήρα του κενού.
  - Ξεκινώντας από το 1, επανάλαβε τόσες φορές όσες το περιεχόμενο της μεταβλητής  $CHARACTERS$  τα παρακάτω βήματα:
    - Εμφάνισε το χαρακτήρα «|» (pipeline).
  - Άλλαξε γραμμή.

## Δραστηριότητα 4: Ταξινόμηση των στοιχείων μονοδιάστατου πίνακα N στοιχείων κατά αύξουσα σειρά με τη μέθοδο «ταξινόμηση φυσαλίδας» (bubble sort)

### Διδακτικοί στόχοι

Στόχος της δραστηριότητας αυτής είναι:

- Να γνωρίσετε μια από τις μεθόδους ταξινόμησης στοιχείων ενός πίνακα, γνωστή με το όνομα *ταξινόμηση φυσαλίδας* (bubble sort), καθώς και την ανάλυσή της σε αλγοριθμικά βήματα.
- Να εξασκηθείτε στη διαχείριση των στοιχείων ενός πίνακα με τη χρήση εμφωλευμένων εντολών επανάληψης.

### Ζητούμενο

Να μελετήσετε τον αλγόριθμο «ταξινόμηση φυσαλίδας» (bubble sort) και στη συνέχεια να κατασκευάσετε τον ψευδοκώδικα που εφαρμόζει τον αλγόριθμο αυτό για την ταξινόμηση των στοιχείων ενός πίνακα A που περιέχει N στοιχεία, όπου N ακέραιος αριθμός ο οποίος εισάγεται από το πληκτρολόγιο κατά την εκτέλεση του προγράμματος. Στην αρχή του προγράμματος εισάγονται στα στοιχεία του πίνακα A τυχαίοι αριθμοί μεταξύ του 1 και του 450. Αποθηκεύστε το πρόγραμμα με το όνομα **Ask\_4.elp**. Εκτελέστε το και δείτε τα αποτελέσματα.

### Υπόδειξη

Στη δραστηριότητα αυτή, για να ταξινομήσουμε τα στοιχεία του πίνακα A, θα πρέπει προηγουμένως, εκτός από τη δημιουργία του, να εκχωρήσουμε (τυχαίες) τιμές στα στοιχεία του.

Για να εκχωρήσουμε σε μια μεταβλητή ή σε στοιχείο ενός πίνακα έναν τυχαίο ακέραιο μεταξύ του 1 και ενός αριθμού X, χρησιμοποιούμε την εντολή:

-Θέσε .. ίσο με τυχαίο αριθμό από 1 έως ..

Στο πρώτο κενό της εντολής εισάγεται η μεταβλητή στην οποία θα εκχωρηθεί ο τυχαίος ακέραιος αριθμός από το 1 έως τον ακέραιο X που εισάγεται στο δεύτερο κενό.

### Περιγραφή της μεθόδου «ταξινόμηση φυσαλίδας»

Στη μέθοδο «ταξινόμηση φυσαλίδας», ξεκινώντας από το πρώτο στοιχείο του πίνακα που θέλουμε να ταξινομήσουμε, σαρώνουμε όλα τα στοιχεία του πίνακα σειριακά, συγκρίνοντας κάθε στοιχείο με το επόμενο. Αν το τρέχον στοιχείο είναι μεγαλύτερο από το επόμενο, τότε τα δύο αυτά στοιχεία ανταλλάσσουν θέση στον πίνακα. Με αυτό τον τρόπο το μεγαλύτερο στοιχείο προωθείται στην τελευταία θέση του πίνακα. Έτσι, την πρώτη φορά θα σαρωθεί ο πίνακας και για τα N στοιχεία του, με αποτέλεσμα στο τέλος της διαδικασίας το μεγαλύτερο από αυτά να καταλήξει στη N-οστή θέση. Τη δεύτερη φορά γίνεται η ίδια διαδικασία για τα πρώτα N-1 στοιχεία του πίνακα, οπότε το μεγαλύτερο από αυτά καταλήγει στη (N-1)-οστή θέση. Την τρίτη φορά επαναλαμβάνεται η διαδικασία αυτή για τα πρώτα N-2 στοιχεία κ.ο.κ. Η διαδικασία συγκρίσεως των στοιχείων ανά δύο συνεχίζεται, μειώνοντας κάθε φορά κατά 1 το πλήθος των στοιχείων του πίνακα που θα σαρωθούν.

Για να ολοκληρωθεί η ταξινόμηση ενός πίνακα N στοιχείων, με τη μέθοδο «ταξινόμηση φυσαλίδας», απαιτούνται συνολικά N-2 επαναλήψεις σάρωσης και

$$\sum_{i=2}^{N-1} i, \text{ συγκρίσεις στοιχείων}$$

**Αλγόριθμος**

- Διάβασε και βάλε στη μεταβλητή  $N$  το πλήθος των στοιχείων που θα ταξινομηθούν.
- Δημιούργησε έναν κενό πίνακα, με όνομα  $A$ ,  $N$  στοιχείων.
- Ξεκινώντας από το 1, επανάλαβε  $N$  φορές τα παρακάτω βήματα:
  - Βάλε στο  $I$ -οστό στοιχείο του πίνακα  $A$  έναν τυχαίο ακέραιο από 1 έως 450, όπου  $I$  ο μετρητής της επαναληπτικής διαδικασίας.
  - Εμφάνισε το  $I$ -οστό στοιχείο του πίνακα  $A$  και στη συνέχεια τους χαρακτήρες κενό+παύλα+κενό, ως διαχωριστικό των αποτελεσμάτων.
- Άφησε δύο κενές γραμμές (για την ευκρίνεια των αποτελεσμάτων).
- Ξεκινώντας από το 1, επανάλαβε  $N-2$  φορές, με μετρητή το  $I$ , τα παρακάτω βήματα:
  - Ξεκινώντας από το 1, επανάλαβε  $N-I$  φορές, με μετρητή το  $J$ , τα παρακάτω βήματα:
    - Συνέκρινε το  $J$ -οστό στοιχείο του πίνακα  $A$ , με το επόμενο  $(J+1)$ -οστό
      - ❖ αν  $A[J] > A[J+1]$  τότε, μετάθεσε τις τιμές τους ως εξής:
        - i. Βάλε στη μεταβλητή  $TEMP$  το περιεχόμενο του στοιχείου  $A[J]$ .
        - ii. Βάλε στο στοιχείο  $A[J]$  το περιεχόμενο του στοιχείου  $A[J+1]$ .
        - iii. Βάλε στο στοιχείο  $A[J+1]$  το περιεχόμενο της μεταβλητής  $TEMP$ .
- Ξεκινώντας από το 1, επανάλαβε  $N$  φορές:
  - Εμφάνισε το  $I$ -οστό στοιχείο του πίνακα  $A$  και στη συνέχεια κενό+παύλα+κενό, ως διαχωριστικό των αποτελεσμάτων. Όπου  $I$  ο μετρητής της επαναληπτικής διαδικασίας.