

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

ΕΠΙΛΟΓΕΣ

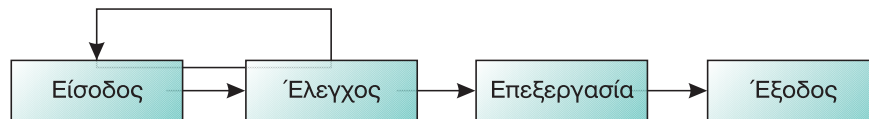
ΑΠΟ ΤΟ ΤΕΤΡΑΔΙΟ

ΜΑΘΗΤΗ

Επιμέλεια : jgalano

“Δίδονται οι βαθμολογίες όλων των μαθητών Γ’ Λυκείου Τεχνολογικής Κατεύθυνσης του σχολικού έτους 1999/2000 στα τέσσερα μαθήματα ειδικότητας. Ζητείται να εκδοθούν στατιστικά αποτελέσματα κατά μάθημα, που περιλαμβάνουν (α) πίνακα συχνοτήτων, (β) τη μέση τιμή και την τυπική απόκλιση”.

Οι απαιτούμενες ενέργειες για την αντιμετώπιση του προβλήματος είναι αυτές που φαίνονται στο σχήμα 1.1.



Σχ. 1.1.

- ⇒ Καταχώριση δεδομένων. Οι βαθμολογίες όλων των μαθητών για ένα μάθημα συγκεντρώνονται και καταγράφονται.
- ⇒ Έλεγχος δεδομένων. Τα δεδομένα ελέγχονται ως προς την ορθότητά τους και γίνονται οι απαραίτητες διορθώσεις, αν απαιτείται.
- ⇒ Επεξεργασία δεδομένων. Γίνονται οι απαραίτητοι υπολογισμοί προκειμένου να βρεθούν τα ζητούμενα αποτελέσματα.
- ⇒ Εξαγωγή αποτελεσμάτων. Δημιουργείται ο πίνακας συχνοτήτων (βλέπε παρ. 1.4 του βιβλίου), σχεδιάζεται το γράφημα και αποτυπώνεται η μέση τιμή και η τυπική απόκλιση.

Από τα παραπάνω μέρη του προβλήματος δεν είναι αρκετά σαφές τι περιλαμβάνει η επεξεργασία δεδομένων. Δηλαδή ποιοι ακριβώς είναι οι απαραίτητοι υπολογισμοί για την εύρεση των αποτελεσμάτων. Έτσι το μέρος αυτό κρίνεται ότι πρέπει να αναλυθεί περισσότερο, όπως στη συνέχεια.

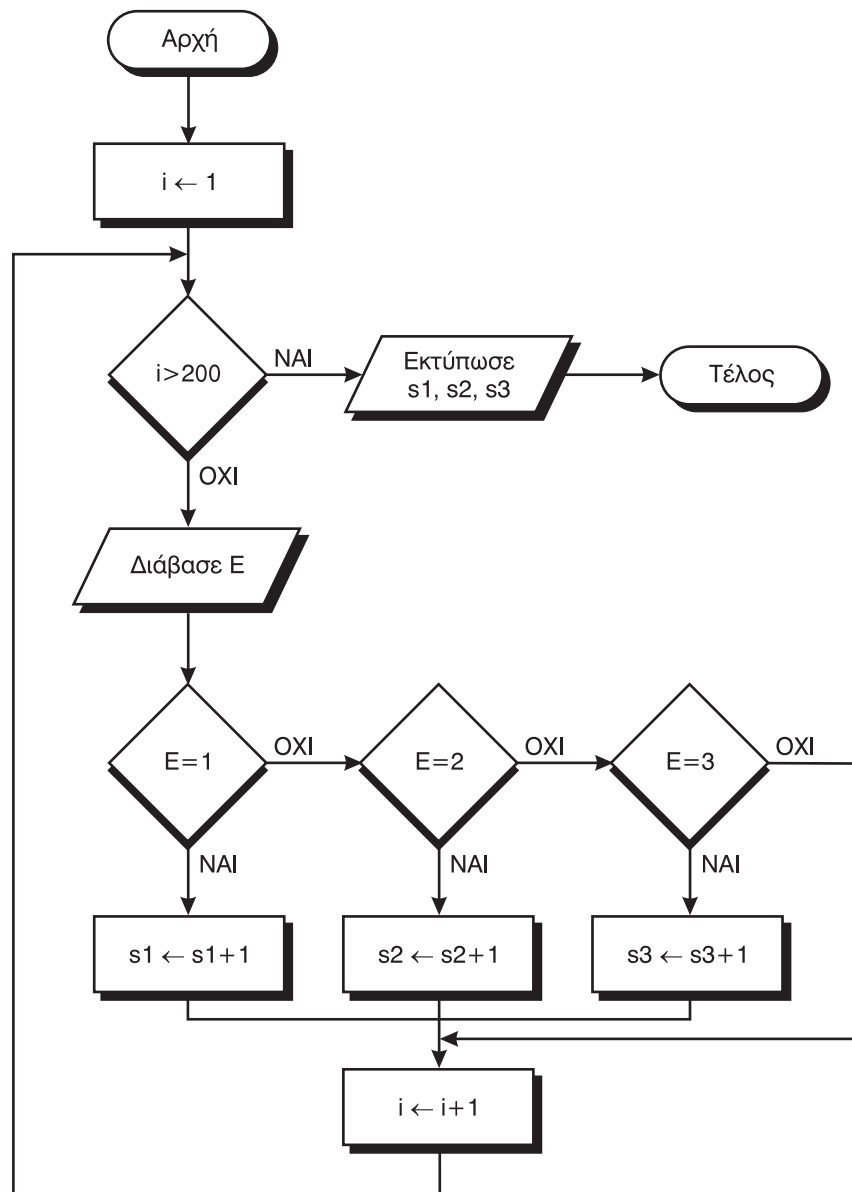
Οι απαιτούμενοι υπολογισμοί είναι:

1. Βρίσκεται το πλήθος όλων των μαθητών, έστω N .
2. Καταμετρείται το πλήθος των μαθητών που έχει βαθμολογία ίση ή μικρότερη του 9 έστω $K1$, από 10 έως 13 έστω $K2$, κ.ο.κ.
3. Το ποσοστό των απορριπτόμενων μαθητών βρίσκεται από τον τύπο $K1/N \cdot 100$.
4. Αθροίζονται όλες οι βαθμολογίες και έστω S το άθροισμα. Η μέση τιμή μ υπολογίζεται από τη σχέση

$$\mu = \frac{S}{N}$$

5. Αθροίζονται επίσης τα τετράγωνα των βαθμολογιών και έστω $S2$ το άθροισμα αυτό. Η τυπική απόκλιση σ βρίσκεται από τον τύπο

$$\sigma^2 = \frac{S2}{N} - \mu^2$$

Διάγραμμα ροής**Αλγόριθμος**

```

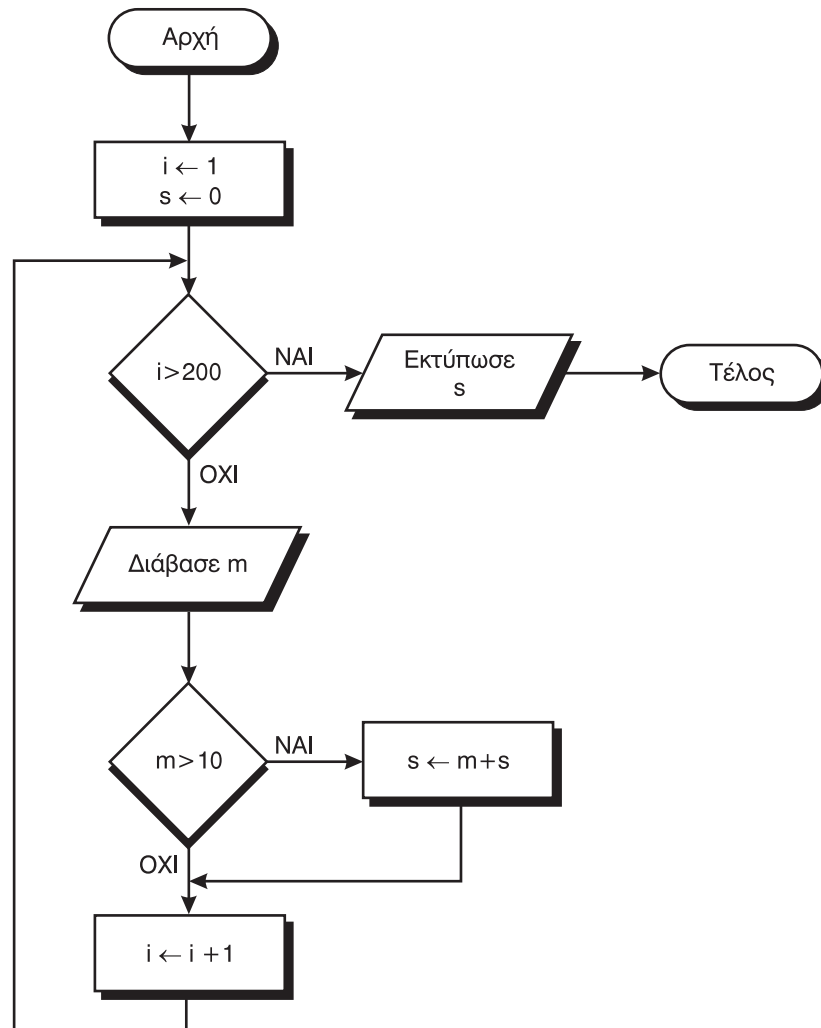
Αλγόριθμος Φοιτητές_Ετος
s1 ← 0 s2 ← 0 s3 ← 0
Για i από 1 μέχρι 200
    Διάβασε E
    Αν E = 1 τότε s1 ← s1 + 1
    αλλιώς_αν E = 2 τότε s2 ← s2 + 1
    αλλιώς_αν E = 3 τότε s3 ← s3 + 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // s1, s2, s3 //
Τέλος Φοιτητές_Ετος
  
```

ΔΤ10. Σε ένα μουσείο υπάρχουν 10 διαφορετικές αίθουσες που περιέχουν διάφορα έργα της ελληνιστικής περιόδου. Κάθε αίθουσα έχει το δικό της αριθμό που είναι από 101, 102, ..., έως 110. Να γράψεις έναν αλγόριθμο που θα διαβάζει τον αριθμό των επισκεπτών κάθε αίθουσας για μία ημέρα και θα υπολογίζει το μέσο όρο των επισκεπτών από όλες τις αίθουσες. Στη συνέχεια ο αλγόριθμος θα πρέπει να εκτυπώνει τους αριθμούς των αιθουσών που είχαν περισσότερους επισκέπτες από το μέσο όρο των επισκεπτών.

Στο σπίτι

Στο τετράδιο σας αντιμετωπίστε τα παρακάτω προβλήματα :

ΔΣ1. Δίνεται το παρακάτω διάγραμμα ροής :



Να δώσετε την εκφώνηση του προβλήματος που εκφράζεται με το συγκεκριμένο διάγραμμα ροής.

ΔΣ4. Δίνεται ο παρακάτω αλγόριθμος :

```

Αλγόριθμος  Ελεγχος_Ανάθεσης
Διάβασε x
Όσο x > 1 επανάλαβε
    Αν x είναι άρτιος τότε
        x ← x/2
    αλλιώς
        x ← 3*x+1
Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα  // x //
Τέλος Ελεγχος_Ανάθεσης
  
```

παρατηρείς ;

Να γράψεις τα αποτελέσματα αυτού του αλγορίθμου για $x=13$, $x=9$ και $x=22$. Τι

Παράδειγμα 2. Υπολογισμός αριθμού συνδυασμών

Είναι γνωστό από τα μαθηματικά ότι ο αριθμός των συνδυασμών των n πραγμάτων ανά k δίνεται από τον τύπο

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Να προταθεί ένας αλγόριθμος για τον υπολογισμό του αριθμού των συνδυασμών αυτών. Ένας απλός τρόπος είναι να εφαρμοσθεί ο προηγούμενος μαθηματικός τύπος. Έτσι δεδομένων των παραλλαγών του αλγορίθμου για τον υπολογισμό του $n!$, προκύπτει ο επόμενος αλγόριθμος που καλεί κάποια από αυτές τις παραλλαγές.

Αλγόριθμος Συνδυασμός

Διάβασε n

Διάβασε k

$a \leftarrow \text{Παραγοντικό}(n)$

$b \leftarrow \text{Παραγοντικό}(k)$

$c \leftarrow \text{Παραγοντικό}(n-k)$

$\text{combination} \leftarrow a / (b * c)$

Γράψε combination

Τέλος Συνδυασμός

Αν και ο αλγόριθμος αυτός είναι ιδιαίτερα απλός στην κατανόηση και στον προγραμματισμό του, εντούτοις δεν είναι ο καλύτερος δυνατός από την άποψη της αποτελεσματικότητας γιατί εκτελεί περιττούς πολλαπλασιασμούς. Αυτό φαίνεται ανάγλυφα αν θεωρήσουμε και πάλι το μαθηματικό τύπο του αριθμού των συνδυασμών. Στο κλάσμα του τύπου αυτού μπορεί να γίνει κάποια απλοποίηση μεταξύ αριθμητή και παρονομαστή. Για παράδειγμα, για τον υπολογισμό του αριθμού των συνδυασμών των 10 πραγμάτων ανά 5, με τον προηγούμενο τρόπο θα εκτελέσουμε 9 πολλαπλασιασμούς για τον υπολογισμό του αριθμητή ($10!$), και $4+4$ πολλαπλασιασμούς για τον υπολογισμό του παρονομαστή (δύο φορές το $5!$). Ενώ εύκολα προκύπτει ότι

$$\frac{10!}{5!5!} = \frac{10 \cdot 9 \cdot 8 \cdot 7 \cdot 6}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}$$

όπου εκτελούνται από τέσσερις πολλαπλασιασμοί σε αριθμητή και παρονομαστή. Ο αντίστοιχος αλγόριθμος έχει ως εξής.

Αλγόριθμος Συνδυασμός2

Διάβασε n

Διάβασε k

$a \leftarrow 1$

Για i **από** n **μέχρι** $n-k+1$ **με_βήμα** -1

$a \leftarrow a * i$

Τέλος_επανάληψης

$b \leftarrow \text{Παραγοντικό}(k)$

$\text{combination} \leftarrow a / b$

Εκτύπωσε combination

Τέλος Συνδυασμός2

Παράδειγμα 5. Αραιοί πίνακες

Ένας πίνακας λέγεται **αραιός** (sparse) αν ένα μεγάλο ποσοστό των στοιχείων του έχουν μηδενική τιμή. Δεν υπάρχει ακριβές ποσοστό σε σχέση με τον αριθμό των μηδενικών στοιχείων, επάνω από το οποίο ένας πίνακας χαρακτηρίζεται ως αραιός. Αρκεί όμως, για παράδειγμα, να πούμε ότι με περισσότερο από 80% μηδενικά ένας πίνακας χαρακτηρίζεται ως αραιός.

Αραιοί πίνακες συναντώνται συχνά σε μεγάλα επιστημονικά προβλήματα (επίλυση εξισώσεων κλπ). Το πρόβλημα με τη διαχείριση των αραιών πινάκων είναι ότι απαιτείται πολύ χώρος για την αποθήκευση μηδενικών. Άρα πρέπει να βρεθεί ένας οικονομικός τρόπος αποθήκευσης των αραιών πινάκων. Στην πράξη έχουν προταθεί αρκετοί τρόποι. Ένας από αυτούς τους τρόπους περιγράφεται στη συνέχεια. Έστω, λοιπόν, ότι δίνεται ο επόμενος πίνακας, που θέλουμε να τον διαχειρισθούμε ως αραιό.

0	7	0	0	0
1	2	0	0	-3
0	0	4	0	0
12	0	0	0	0

Αντί να αποθηκεύσουμε αυτόν το διδιάστατο πίνακα 4x5, θα θεωρήσουμε ένα μονοδιάστατο πίνακα όπου θα τοποθετήσουμε μόνο τα μη μηδενικά στοιχεία, για τα οποία όμως χρειαζόμαστε τα στοιχεία των αντίστοιχων γραμμών και στηλών. Έτσι καταλήγουμε κάθε μη μηδενικό στοιχείο να αντιπροσωπεύεται από μία τριάδα στοιχείων, δηλαδή <γραμμή,στήλη,τιμή>. Για το λόγο αυτό δημιουργούμε ένα μονοδιάστατο πίνακα 18 θέσεων για τα 6 μη μηδενικά στοιχεία του αρχικού πίνακα. Ο νέος πίνακας έχει τη μορφή

1	2	7	2	1	1	2	2	2	2	5	-3	3	3	4	4	1	12
---	---	---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----

Πλέον, το πρόβλημα έγκειται στην αναγνώριση της τιμής μίας θέσης του παλαιού πίνακα, δεδομένου ότι ο πίνακας είναι αποθηκευμένος με τη νέα του μορφή. Ο επόμενος αλγόριθμος "Αραιός" επιστρέφει την τιμή του στοιχείου που βρίσκεται στη θέση <γραμμή l, στήλη m> του αρχικού πίνακα επεξεργαζόμενος τη νέα μορφή του πίνακα που αποτελείται από 3n θέσεις, όπου n ο αριθμός των μη μηδενικών στοιχείων.

Αλγόριθμος Αραιός

Δεδομένα // sparse, n //

flag ← 0

k ← 0

Οσο flag=0 **επανάλαβε**

 i ← sparse[3*k+1]

 j ← sparse[3*k+2]

Αν i=L **και** j=M **τότε**

 result ← sparse[3*k+3]

 flag ← 1

αλλιώς_αν i > L **ή** (i=L **και** j > M) **τότε**

 result ← 0

 flag ← 1

αλλιώς

 k ← k+1

Τέλος_αν

Τέλος_επανάληψης

Αποτελέσματα // result //

Τέλος Αραιός



ΔΤ2. Ο αλγόριθμος της φουσσαλίδας όπως διατυπώθηκε στην παράγραφο 3.7 έχει το μειονέκτημα ότι δεν είναι αρκετά “έξυπνος” ώστε να διαπιστώνει στην αρχή ή στο μέσο της διαδικασίας αν ο πίνακας είναι ταξινομημένος. Να σχεδιασθεί μία παραλλαγή του αλγορίθμου αυτού που να σταματά όταν διαπιστωθεί ότι τα στοιχεία του πίνακα είναι ήδη ταξινομημένα.

Υπόδειξη: Να χρησιμοποιήσετε μία βοηθητική μεταβλητή που να ελέγχει το τέλος κάθε επανάληψης του εξωτερικού βρόχου (“Για i από 2 μέχρι n ”) αν για την τρέχουσα τιμή του i έγιναν αντιμεταθέσεις στοιχείων.



ΔΣ3. Εστω ότι θέλουμε να διατάξουμε τους μαθητές μίας τάξης κατά φθίνουσα σειρά ύψους. Η τεχνική που θα ακολουθήσουμε είναι η εξής. Αρχικά, τοποθετούμε τους μαθητές σε μία τυχαία σειρά. Κατόπιν συγκρίνουμε το δεύτερο με τον πρώτο και αν χρειασθεί τους αντιμεταθέτουμε ώστε πρώτος να είναι ο ψηλότερος. Στη συνέχεια θεωρούμε τον τρίτο και τον τοποθετούμε στη σωστή σειρά σε σχέση μεν πρώτο και το δεύτερο. Κατ’ αυτόν τον τρόπο συνεχίζουμε μέχρι να τοποθετήσουμε στη σωστή σειρά όλους τους μαθητές. Να σχεδιασθεί ένας αλγόριθμος που να υλοποιεί αυτή τη μέθοδο ταξινόμησης.

Παράδειγμα 3. Εύρεση δύο μικρότερων αριθμών.

Σε ένα Τμήμα μίας επιχείρησης χρειάζεται να βρεθούν οι δύο χαμηλότεροι μισθοί με δεδομένο ότι το Τμήμα απασχολεί 50 υπαλλήλους και οι μισθοί τους αποθηκεύονται σε κάποιον πίνακα. Να γραφεί ένας αλγόριθμος που θα υπολογίζει τους δύο μικρότερους μισθούς με δεδομένο τον πίνακα των μισθών των υπαλλήλων.



Το πρόβλημα της ανεύρεσης των δύο μικρότερων στοιχείων ενός πίνακα επιδέχεται διάφορες τεχνικές και τρόπους σχεδίασης. Ο αλγόριθμος που παρουσιάζεται στη συνέχεια είναι αρκετά απλός και δεν έχει καλή αποδοτικότητα.

```
Αλγόριθμος Δύο_Μικρότεροι
Δεδομένα // M //
low1 ← M[1]
pos ← 1
Για i από 2 μέχρι 50
    Αν M[i] < low1 τότε
        low1 ← M[i]
        pos ← i
Τέλος_αν
Τέλος_επανάληψης

Αν pos <> 1 τότε
    low2 ← M[1]
αλλιώς
    low2 ← M[2]
Τέλος_αν
Για i από 2 μέχρι 50
    Αν (i <> pos και M[i] < low2) τότε
        low2 ← M[i]
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // low1 , low2 //
Τέλος Δύο_Μικρότεροι
```

Παράδειγμα 3. Ανάλυση αλγορίθμου ταξινόμησης.

Έστω ότι έχουμε την παρακάτω απλή μορφή για τον αλγόριθμο ταξινόμησης με ευθεία ανταλλαγή (bubblesort) :

```
Αλγόριθμος Ευθεία_Ανταλλαγή
Δεδομένα // A //
Για i από 1 μέχρι n-1
    Για j από 1 μέχρι n-1
        Αν A[j+1] < A[j] τότε
            Αντιμετάθεσε A[j+1], A[j]
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτελέσματα // A //
```

```
Τέλος Ευθεία_Ανταλλαγή
```

Έστω ότι έχουμε τον πίνακα A με τα παρακάτω στοιχεία :

1	2	3	4
7	4	3	8

Να παρακολουθήσετε την πορεία του αλγορίθμου με καταγραφή της επίδοσής του που θα εκφράζεται από τον αριθμό των πράξεων που πρέπει να εκτελεσθούν.

ΔΣ4. Έστω ότι έχεις τον παρακάτω αλγόριθμο :

```
Αλγόριθμος Ευθεία_Ανταλλαγή2
Δεδομένα // A //
Για i από 1 μέχρι n-1
    Για j από 1 μέχρι n-1
        Αν A[j+1] < A[j] τότε
            Αντιμετάθεσε A[j+1], A[j]
        Τέλος_αν
    Τέλος_επανάληψης
    Εκτύπωσε A[j]
Τέλος_επανάληψης
Αποτελέσματα // A //
```

```
Τέλος Ευθεία_Ανταλλαγή2
```

Να σχολιάσεις την πολυπλοκότητα του αλγορίθμου. Χρησιμοποίησε το Παράδειγμα 2 και να ελέγξεις αν ο παραπάνω αλγόριθμος είναι ίδιας ή διαφορετικής πολυπλοκότητας από τον αλγόριθμο του Παραδείγματος 2.

6.2. Δραστηριότητες - ασκήσεις



Στην τάξη

ΔΤ1. Ο παρακάτω αλγόριθμος αποτελεί τμήμα μη δομημένου προγράμματος.

Να γράψεις αλγόριθμο σχεδιασμένο με τις αρχές του δομημένου προγραμματισμού, που να εκτελεί τις ίδιες λειτουργίες.

```
ΑΡΧΗ
ΟΣΟ συνθήκη1 ΕΠΑΝΑΛΑΒΕ
    Εντολή 2
    ΑΝ συνθήκη3 ΤΟΤΕ
        Εντολή4
        Πήγαινε στο Τέλος
    ΑΛΛΙΩΣ
        Εντολή5
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ
```

ΔΣ2. Ο παρακάτω αλγόριθμος αποτελεί τμήμα μη δομημένου προγράμματος.

Να γράψεις αλγόριθμο σχεδιασμένο με τις αρχές του δομημένου προγραμματισμού, που να εκτελεί τις ίδιες λειτουργίες.

```
ΑΡΧΗ
ΑΝ συνθήκη1 ΤΟΤΕ
    Εντολή1
    ΑΝ συνθήκη2 ΤΟΤΕ
        Εντολή2
        Εντολή3
        Πήγαινε στην Εντολή5
    ΤΕΛΟΣ_ΑΝ
    Εντολή4
    Εντολή5
    Πήγαινε στην Αρχή
ΤΕΛΟΣ_ΑΝ
Εντολή3
ΤΕΛΟΣ
```

ΔΤ2. Τι τύπου μεταβλητές πρέπει να χρησιμοποιήσετε για τα παρακάτω στοιχεία του μαθητολόγιου του σχολείου μας; Γράψετε το αντίστοιχο τμήμα δηλώσεων.

1. Το όνομα ενός μαθητή.
2. Ο αριθμός μαθητολογίου του μαθητή.
3. Τη βαθμολογία του μαθητή.
4. Το τηλέφωνο ενός μαθητή.
5. Τη διεύθυνση ενός μαθητή.
6. Το φύλο ενός μαθητή (πώς μπορεί να οριστεί με χρήση λογικής μεταβλητής;)

8.3. Συμβουλές - υποδείξεις



Εφόσον όπως έχουμε αναφέρει πολλές φορές κάθε πρόγραμμα μπορεί να υλοποιηθεί με τη χρήση των τριών δομών της ακολουθίας, της επιλογής και της επανάληψης, αν μάθεις να χρησιμοποιείς σωστά τις εντολές επιλογής και επανάληψης, μπορείς να υλοποιήσεις σχεδόν οποιονδήποτε αλγόριθμο. Στην πραγματικότητα όμως μόνο η εξάσκηση και η πείρα θα σου εξασφαλίσουν τη δυνατότητα να συντάσσεις εύκολα και γρήγορα σωστά προγράμματα. Οι παρακάτω συμβουλές θα σε βοηθήσουν στη συγγραφή σωστών προγραμμάτων αποφεύγοντας μερικά από τα πιο συνηθισμένα λάθη που παρουσιάζονται.

- ⇒ Όταν χρησιμοποιείς σύνθετες λογικές εκφράσεις, να προσέχεις την ιεραρχία των τελεστών. Είναι καλύτερο να χρησιμοποιείς πάντα παρενθέσεις, έστω και αν δεν είναι απαραίτητο, σε προφυλάσσει από πιθανά λάθη και αβλεψίες, ενώ ταυτόχρονα κάνει το πρόγραμμα πιο εύκολο στην κατανόηση του.
- ⇒ Πριν χρησιμοποιήσεις εμφωλευμένα ΑΝ, σκέψου μήπως το ίδιο πρόγραμμα μπορεί να υλοποιηθεί απλούστερα με σύνθετες λογικές εκφράσεις, την εντολή ΑΝ-ΑΛΛΙΩΣ_ΑΝ ή κάποια άλλη εντολή επιλογής που πιθανόν να προσφέρει το υπολογιστικό περιβάλλον που χρησιμοποιείς.
- ⇒ Οι μεταβλητές που ελέγχουν την επανάληψη του βρόχου ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ πρέπει υποχρεωτικά να αλλάζουν τιμή μέσα στο σώμα του βρόχου, αλλιώς ή δεν εκτελείται ποτέ ή συνηθέστερα δεν σταματάει η εκτέλεση του (ατέρμων βρόχος).
- ⇒ Οι επαναλήψεις που υλοποιούνται με την εντολή ΟΣΟ, μπορεί να μην εκτελεστούν ούτε μία φορά, αφού ο έλεγχος γίνεται στην είσοδο του βρόχου, αντίθετα οι επαναλήψεις ΜΕΧΡΙΣ_ΟΤΟΥ θα πραγματοποιηθούν τουλάχιστον μία φορά.
- ⇒ Η εντολή ΓΙΑ χρησιμοποιείται μόνο για προκαθορισμένο αριθμό επαναλήψεων. Αν λοιπόν ξέρεις τον αριθμό των επαναλήψεων ή μπορείς να τον υπολογίσεις, τότε να χρησιμοποιείς την εντολή ΓΙΑ.
- ⇒ Ποτέ μη χρησιμοποιείς εντολές που αλλάζουν την αρχική τιμή, την τελική τιμή, το βήμα ή τη μεταβλητή που ελέγχει την επανάληψη μέσα σε ένα βρόχο ΓΙΑ. Αν και μερικές γλώσσες προγραμματισμού επιτρέπουν αυτές τις αλλαγές, να τις αποφεύγεις, γιατί οδηγούν σε προγράμματα δυσνόητα και συνήθως λανθασμένα.

ΔΤ5. Διάβασε προσεκτικά τα παρακάτω τμήματα προγράμματος. Ποια είναι τα λάθη; Διόρθωσέ τα, ώστε να λειτουργούν σωστά.

A.

```
ΔΙΑΒΑΣΕ Μισθός
ΟΣΟ Μισθός <> 0 ΕΠΑΝΑΛΑΒΕ
    Αθροισμα <- 0
    ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Αθροισμα <- Αθροισμα+Μισθός
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B.

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    Αθροισμα <- 0
    ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Αθροισμα <- Αθροισμα+Μισθός
    ΔΙΑΒΑΣΕ Μισθός
ΜΕΧΡΙΣ_ΟΤΟΥ Μισθός<>0
```

Γ.

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
  Αθροισμα <- 0
  ΔΙΑΒΑΣΕ Μισθός
  ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
    Μέγιστος <- Μισθός
  ΤΕΛΟΣ_ΑΝ
  ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
    Ελάχιστος <- Μισθός
  ΤΕΛΟΣ_ΑΝ
  Αθροισμα <- Αθροισμα+Μισθός
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Εκτέλεσε εικονικά τις εντολές στο χαρτί και σημείωνε τα αποτελέσματα που προκύπτουν. Με αυτόν τον τρόπο θα δεις τα λάθη και στη συνέχεια θα κάνεις τις διορθώσεις.

ΔΕ1. Να γραφεί πρόγραμμα που να διαβάζει το βαθμό ενός μαθητή και να υπολογίζει την αντίστοιχη αξιολόγηση του με βάση το βαθμό του και σύμφωνα με τον παρακάτω πίνακα:

17,5 -20	Άριστα
15,5 -17,4	Πολύ καλά
13,5 - 15,4	Καλά
9,5 - 13,4	Μέτρια
0 - 9,4	Απορρίπτεται

Το πρόγραμμα να γραφεί με τους ακόλουθους τρόπους:

- ⇒ Με εντολές ΑΝ ... ΤΟΤΕ
- ⇒ Με εντολές ΑΝ ... ΤΟΤΕ ... ΑΛΛΙΩΣ_ΑΝ
- ⇒ Με εμφωλευμένα ΑΝ.
- ⇒ Με την εντολή ΕΠΙΛΕΞΕ

ΔΕ3. Να γραφεί πρόγραμμα το οποίο θα εκτελεί κάποια από τις βασικές πράξεις πρόσθεσης, αφαίρεσης, πολλαπλασιασμό και διαίρεσης ανάμεσα σε δύο ακέραιους αριθμούς και θα εμφανίζει το αποτέλεσμα στην οθόνη.

Το πρόγραμμα θα ελέγχεται από το παρακάτω μενού επιλογής και θα σταματάει όταν ο χρήστης επιλέξει από το μενού την επιλογή έξοδο.

1. Πρόσθεση
2. Αφαίρεση
3. Πολλαπλασιασμό
4. Διαίρεση
5. Έξοδος

Δώσε επιλογή: __

Διάλεξε ένα μεταξύ των προτεινόμενων

8. Ποιο από τα παρακάτω υπολογίζει το άθροισμα των 100 πρώτων περιττών αριθμών

A.

```
Αθροισμα <- 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    Αθροισμα <- Αθροισμα+Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B.

```
Αθροισμα <- 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
    Αθροισμα <- Αθροισμα+ Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ.

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
    Αθροισμα <- 0
    Αθροισμα <- Αθροισμα+ Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δ.

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
    Αθροισμα <- Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

9. Τι θα εκτυπώσει το παρακάτω τμήμα προγράμματος

```
A <- 0
ΓΙΑ Ι ΑΠΟ 10 ΜΕΧΡΙ 20 ΜΕ_ΒΗΜΑ 10
    A <- A+I^2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ A
```

A. 0 B. 100 Γ. 500 Δ. 400

10. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    A <- 0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
        A <- A-1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΕΧΡΙΣ_ΟΤΟΥ A=0
```

A. 10 B. 0 Γ. 5 Δ. Άπειρες

11. Δίνονται οι παρακάτω εντολές

```
A <- 1
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2
    A <- A*I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Ποιες από τις επόμενες ομάδες εντολών δίνουν στο A την ίδια τιμή

A.

```
A <- 1
I <- 1
ΟΣΟ I<=10 ΕΠΑΝΑΛΑΒΕ
    I <- I+2
    A <- A*I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B.

```
A <- 1
I <- 1
ΟΣΟ I <=10 ΕΠΑΝΑΛΑΒΕ
    A <- A*I
    I <- I+2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ.

```
A <- 1
I <- 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  A <- A*I
  I <- I+2
ΜΕΧΡΙΣ_ΟΤΟΥ I<10
```

Δ.

```
A <- 1
I <- 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  A <- A*I
  I <- I+2
ΜΕΧΡΙΣ_ΟΤΟΥ I=10
```

12. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ 3
  ΓΡΑΨΕ 'Μήνυμα'
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

A. 2 B. 0 Γ. 1 Δ. Άπειρες

13. Ποια η λειτουργία του παρακάτω τμήματος προγράμματος

```
B <- 10
ΔΙΑΒΑΣΕ A
B <- A
ΑΝ A < 0 ΤΟΤΕ
  B <- -A
ΤΕΛΟΣ_ΑΝ
A <- 0
ΓΡΑΨΕ B
```

- A. Τυπώνει τον αριθμό που διάβασε
- B. Τυπώνει την απόλυτη τιμή του αριθμού που διάβασε
- Γ. Τυπώνει πάντα την τιμή 0
- Δ. Τυπώνει πάντα την τιμή 10

Παράδειγμα 3

Δίνονται δύο ταξινομημένοι κατά αύξουσα σειρά μονοδιάστατοι πίνακες, ακεραίων αριθμών. Να γραφεί πρόγραμμα το οποίο να συγχωνεύει τους δύο πίνακες σε ένα τρίτο ο οποίος να είναι επίσης ταξινομημένος κατά αύξουσα σειρά. Οι δύο αρχικοί πίνακες δεν μπορούν να περιέχουν περισσότερα από 100 στοιχεία ο καθένας.

Η συγχώνευση είναι μία βασική λειτουργία των πινάκων και γενικότερα των δομών δεδομένων. Στη συνέχεια δίνεται ένας πολύ απλός αλγόριθμος συγχώνευσης δύο ταξινομημένων πινάκων σε ένα τρίτο ταξινομημένο πίνακα.

Θεωρείται ότι στην είσοδο του αλγορίθμου συγχώνευσης δίνονται δύο ταξινομημένοι, κατά αύξουσα σειρά, πίνακες A και B, μεγέθους N και M στοιχείων αντίστοιχα, ενώ στην έξοδο προκύπτει ένας τρίτος πίνακας Γ με N+M ταξινομημένα στοιχεία επίσης κατά αύξουσα σειρά.

Στο πρόγραμμα Συγχώνευση που ακολουθεί οι μεταβλητές i, j και k είναι δείκτες για την κίνηση μέσα στους πίνακες A, B και Γ. Η μέθοδος προχωρεί ως εξής:

Το μικρότερο στοιχείο από τους πίνακες A και B τοποθετείται στον πίνακα Γ με ταυτόχρονη αύξηση του αντίστοιχου δείκτη. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου τελειώσουν τα στοιχεία του ενός πίνακα.

Στη συνέχεια τα υπόλοιπα στοιχεία του άλλου πίνακα μεταφέρονται στον πίνακα Γ.

ΠΡΟΓΡΑΜΜΑ Συγχώνευση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:A[100], B[100], Γ[200], I, J, K, N, M, Λ
! A και B αρχικοί πίνακες
! Γ τελικός πίνακας

ΑΡΧΗ

! Διάβασε τα δεδομένα

ΓΡΑΨΕ 'Δώσε το πλήθος των στοιχείων του πίνακα A (<100)'

ΔΙΑΒΑΣΕ N

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΔΙΑΒΑΣΕ A[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε το πλήθος των στοιχείων του πίνακα B(<100)'

ΔΙΑΒΑΣΕ M

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** M

ΔΙΑΒΑΣΕ B[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Συγχώνευση πινάκων

! I είναι ο δείκτης για τον πίνακα A

! J είναι ο δείκτης για τον πίνακα B

! K είναι ο δείκτης για τον πίνακα Γ

I <- 1

J <- 1

K <- 1

ΟΣΟ I <= N **ΚΑΙ** J <= M **ΕΠΑΝΑΛΑΒΕ**

! Όσο και τα δύο έχουν στοιχεία

ΑΝ A[I] < B[J] **ΤΟΤΕ**

 Γ[K] <- A[I]

 K <- K+1

 I <- I+1

ΑΛΛΙΩΣ

 Γ[K] <- B[J]

 K <- K+1

 J <- J +1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Μεταφορά των υπολοίπων στοιχείων του A ή του B

ΑΝ I > N **ΤΟΤΕ**

ΓΙΑ Λ **ΑΠΟ** K **ΜΕΧΡΙ** N+M

 Γ[Λ] <- B[J]

 J <- J +1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΛΙΩΣ

ΓΙΑ Λ **ΑΠΟ** K **ΜΕΧΡΙ** N+M

 Γ[Λ] <- A[I]

 I <- I+1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Εκτύπωση τελικού πίνακα

ΓΙΑ Λ **ΑΠΟ** 1 **ΜΕΧΡΙ** N+M

ΓΡΑΨΕ Γ[Λ]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Συγχώνευση

ΔΤ3. Να γράψετε τις εντολές που δίνουν τις ακόλουθες τιμές σε ένα πίνακα ακεραίων A.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

ΔΤ4. Να γραφούν οι εντολές που ανταλλάσσουν τα στοιχεία της τρίτης και της έκτης στήλης σε ένα πίνακα ακεραίων 5X6.

ΔΕ1. Να γράψετε ένα πρόγραμμα το οποίο να διαβάζει των αριθμό των τερμάτων που σημειώθηκαν στους αγώνες ποδοσφαίρου μίας αγωνιστικής της Α κατηγορίας (9 τιμές), να υπολογίζει τον μέσο αριθμό τερμάτων καθώς και το εύρος των τερμάτων (δηλαδή τη διαφορά της μεγαλύτερης από την μικρότερη τιμή).

ΔΣ6. Δίνονται οι πίνακες Σ1(K,K) και Π1(K,K) που περιέχουν τα αποτελέσματα των αγώνων ομίλου του EuroBasket. Ο πίνακας Σ1 περιέχει τα αποτελέσματα των αγώνων (N (νίκη) ή H (ήττα)), ενώ ο πίνακας Π1 τη διαφορά πόντων για κάθε αγώνα.

Να γραφεί πρόγραμμα το οποίο θα βρίσκει και θα εκτυπώνει την τελική βαθμολογία του ομίλου. Σε περίπτωση ισοβαθείας προηγείται η ομάδα που έχει την καλύτερη διαφορά πόντων από τις ισόβαθμες της.

Τα στοιχεία της κύριας διαγωνίου δεν περιέχουν καμία πληροφορία (καμία ομάδα δεν παίζει με τον εαυτό της!).

Ο πίνακας περιέχει στοιχεία μόνο κάτω ή πάνω από τη διαγώνιο του, είναι δηλαδή τριγωνικός (κάθε ομάδα παίζει μόνο μία φορά με κάθε αντίπαλο).

10.2. Επιπλέον παραδείγματα

Παράδειγμα 1

Πολλά από τα προγράμματα που αναπτύχθηκαν στα προηγούμενα κεφάλαια μπορούν να γραφούν καλύτερα με τη χρήση υποπρογραμμάτων. Εδώ θα δούμε το πρόγραμμα που υπολογίζει τα βασικά στατιστικά μεγέθη τη μέση τιμή, την τυπική απόκλιση και τη διάμεσο τιμή που παρουσιάστηκε στο βιβλίο σου στο κεφάλαιο 9.

Το πρόγραμμα χρησιμοποιεί τις εξής διαδικασίες και συναρτήσεις:

Υπολόγισε_ΜΟ_ΤυπΑπ :Υπολογίζει τη μέση τιμή και την τυπική απόκλιση ακεραίων αριθμών. Το τμήμα αυτό θα μπορούσε να υλοποιηθεί και με δύο συναρτήσεις, μία για τον υπολογισμό της μέσης τιμής και μίας δεύτερης για τον υπολογισμό της τυπικής απόκλισης.

Ταξινόμηση: Η διαδικασία αυτή ταξινομεί τα στοιχεία του πίνακα χρησιμοποιώντας μία παραλλαγή του αλγορίθμου που παρουσιάστηκε στο βιβλίο σου.

Υπολογισμός_Διαμέσου: Πραγματική συνάρτηση η οποία υπολογίζει τη διάμεσο τιμή.

ΠΡΟΓΡΑΜΜΑ Στατιστική

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : I, Πλήθος, Στοιχεία[100], Μέγιστο, Διάμεσος,
Αθροισμα, Αθροισμα_2, Βοηθητική

ΠΡΑΓΜΑΤΙΚΕΣ: MO, Τυπ_Απόκλιση

ΑΡΧΗ

! Εισαγωγή στοιχείων

ΓΡΑΨΕ 'Δώσε το πλήθος των αριθμών (μέγιστο 100)'

ΔΙΑΒΑΣΕ Πλήθος

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** Πλήθος

ΓΡΑΨΕ 'Δώσε έναν αριθμό'

ΔΙΑΒΑΣΕ Στοιχεία[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΚΑΛΕΣΕ Υπολόγισε_MO_ΤυπΑπ(Στοιχεία, Πλήθος, MO, Τυπ_Απόκλιση)

ΚΑΛΕΣΕ Ταξινόμησε(Στοιχεία, Πλήθος)

Διάμεσος <- Υπολογισμός_Διαμέσου(Στοιχεία)

! Εκτύπωση αποτελεσμάτων

ΓΡΑΨΕ 'ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΤΟΥΣ ', Πλήθος, 'ΑΡΙΘΜΟΥΣ'

ΓΡΑΨΕ 'ΜΕΣΟΣ ΟΡΟΣ:', MO

ΓΡΑΨΕ 'ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ: ', Τυπ_Απόκλιση

ΓΡΑΨΕ 'ΔΙΑΜΕΣΟΣ:', Διάμεσος

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Στατιστική

ΔΙΑΔΙΚΑΣΙΑ Υπολόγισε_MO_ΤυπΑπ(Πίνακας, N, MO, ΤυπΑποκλ)

! Υπολογισμός μέσου όρου

!Υπολογισμός τυπικής απόκλισης

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:Πίνακας[100],N, I, Αθροισμα, Αθροισμα_2

ΠΡΑΓΜΑΤΙΚΕΣ:MO, ΤυπΑποκλ

ΑΡΧΗ

Αθροισμα <- 0

Αθροισμα_2 <- 0

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** N

Αθροισμα <- Αθροισμα+ Πίνακας[I]

Αθροισμα_2 <- Αθροισμα_2+ Πίνακας[I]^2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

MO <- Αθροισμα/N

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Υπολόγισε_MO_ΤυπΑπ

ΔΙΑΔΙΚΑΣΙΑ Ταξινόμησε(Πίνακας, N)

!Ταξινόμηση των στοιχείων του πίνακα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:I,N1,T,Βοηθητική

ΑΡΧΗ

N1 <- N

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

T <- 0

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** N1-1

ΑΝ Πίνακας[I] > Πίνακας[I+1] **ΤΟΤΕ**

Βοηθητική <- Πίνακας[I]

Πίνακας[I] <- Πίνακας[I+1]

Πίνακας[I+1] <- Βοηθητική

T <- I

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

N1 <- T

ΜΕΧΡΙΣ_ΟΤΟΥ T=0

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Ταξινόμησε

ΣΥΝΑΡΤΗΣΗ Υπολογισμός_Διαμέσου (A,N) :**ΑΚΕΡΑΙΑ**
ΑΚΕΡΑΙΕΣ:A[100],N

ΑΡΧΗ

ΑΝ N MOD 2 =0 **ΤΟΤΕ**

Υπολογισμός_Διαμέσου <- (A[N/2]+A[N/2+1])/2

ΑΛΛΙΩΣ

Υπολογισμός_Διαμέσου <- A[(N+1)/2]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ Υπολογισμός_Διαμέσου



ΔΣ3. Να επεκτείνεις το παράδειγμα 1, ώστε να υπολογίζει την επικρατούσα τιμή, δηλαδή την τιμή που εμφανίζεται περισσότερες φορές.

14.2. Επιπλέον παραδείγματα



Παράδειγμα 1

Δίνεται ο διδιάστατος πίνακας A(i,k) όπου στην πρώτη διάσταση περιέχει την κωδικό είδους ενός υλικού και στη δεύτερη την ποσότητα του υλικού. Ο πίνακας A είναι ταξινομημένος κατά είδος. Ζητείται να εκτυπώνονται σύνολα ποσότητας, σε κάθε αλλαγή είδους και η συχνότητα εμφάνισης κάθε κωδικού. Στο τέλος της επεξεργασίας να τυπώνεται γενικό σύνολο ποσότητας.

Παρακάτω δίνεται ένας πίνακας με εικονικά δεδομένα και τα αποτελέσματα που πρέπει να παράγει το πρόγραμμά μας με αυτά τα δεδομένα.

Δεδομένα		Αποτελέσματα	
Κωδικός Είδους	Ποσότητα	Σύνολο Είδους	Γενικό Σύνολο
101	20		
101	30		
101	15		
101	20		
101	22	107	
105	1		
105	7	8	
200	17		
200	2		
200	3	22	
250	29	29	
280	12		
280	13	25	
310	6	6	
320	7	7	
330	9	9	213

Το πρόβλημα είναι παρόμοιο με το παράδειγμα 1 του βιβλίου. Εδώ τη θέση του αριθμού παίρνει ο κωδικός είδους. Επομένως θα αναζητήσεις την αλλαγή τιμής στο πεδίο του κωδικού είδους. Αν ακολουθήσεις την λογική της πρώτης λύσης του 1^{ου} παραδείγματος του βιβλίου, θα έχεις το παρακάτω πρόγραμμα σαν λύση του προβλήματος.

```
ΠΡΟΓΡΑΜΜΑ Σύνολα_Είδους
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A, GS, i, S, Προηγ_A
ΑΡΧΗ
    GS <- 0
    i <- 1
    S <- 0
    Προηγ_A <- A(1,1)
    ΟΣΟ i<1001 ΕΠΑΝΑΛΑΒΕ
        ΑΝ Προηγ_A<>A(i,1) ΤΟΤΕ
            ΓΡΑΨΕ Προηγ_A, S
            GS <- GS+S
            Προηγ_A <- A(i,1)
            S <- 0
        ΤΕΛΟΣ ΑΝ
        S <- S+A(i,2)
        i <- i+1
    ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ Προηγ_A, S
    GS <- GS+S
    ΓΡΑΨΕ GS
ΤΕΛΟΣ Σύνολα_Είδους
```

Παράδειγμα 2

Ένας αριθμός λέγεται πρώτος αν διαιρείται μόνο από το 1 και τον εαυτό του. Έστω, λοιπόν, ότι δίνεται ένας θετικός ακέραιος αριθμός n και ζητείται να διαπιστωθεί αν είναι πρώτος ή όχι. Μία πρώτη σκέψη είναι να αρχίσουμε να διαιρούμε τον αριθμό αυτόν διαδοχικά με 2, 3, 4, ..., $n-1$. Έτσι, αν ο αριθμός n δεν διαιρείται ακριβώς με κανένα από τους αριθμούς αυτούς, τότε πράγματι ο αριθμός n είναι πρώτος.

```
Αλγόριθμος ΠρώτοςΑριθμός
Διάβασε n
flag ← Ψευδής
i ← 2
Αρχή_επανάληψης
    Αν  $n \bmod i = 0$  τότε flag ← Αληθής
    i ← i+1
Μέχρις_ότου flag=Αληθής ή  $i > n-1$ 
Αποτελέσματα // flag //
Τέλος ΠρώτοςΑριθμός
```

Ο προηγούμενος αλγόριθμος χρησιμοποιεί μία συνάρτηση “mod” που επιστρέφει το υπόλοιπο της ακεραίας διαίρεσης. Αν αυτό είναι μηδέν τότε ο αριθμός δεν είναι πρώτος και η σημαία flag γίνεται Αληθής.

Με λίγη προσοχή και ελάχιστο κόπο, ο προηγούμενος αλγόριθμος μπορεί να γίνει ταχύτερος αν αλλάξει η συνθήκη της εντολής “Αρχή_επανάληψης ... μέχρις ότου” από $i > n-1$ σε $i > \sqrt{n}$. Αυτό μπορεί να γίνει με βεβαιότητα, καθώς αποδεικνύεται και μαθηματικά. Για παράδειγμα, αν θέλουμε να διαπιστώσουμε αν ο αριθμός 11 είναι πρώτος, τότε δεν είναι απαραίτητο να διαιρέσουμε το 11 δια 2, 3, κλπ μέχρι το 10, αλλά μέχρι το 4. Ο λόγος είναι ότι αν το 11 διαιρείτο ακριβώς δια του 5, του 6 κλπ, τότε θα διαιρείτο ακριβώς και δια του 11/5, 11/6 κλπ, γεγονός που θα είχε ήδη αποκαλυφθεί.



Παράδειγμα 3. Πίνακας Αποφάσεων

Είναι γνωστό ότι δίσεκτο έτος λέγεται αυτό που ο αριθμός του είναι πολλαπλάσιο του 4, αλλά όχι του 100, εκτός αν είναι πολλαπλάσιο του 400. Για παράδειγμα: το 1984 είναι δίσεκτο (πολλαπλάσιο του 4), το 1900 δεν είναι (πολλαπλάσιο του 4 αλλά και του 100), το 1600 είναι δίσεκτο (πολλαπλάσιο του 4, του 100, αλλά και του 400), τέλος το 1993 δεν είναι. Η διατύπωση της λύσης για ένα τέτοιο έλεγχο είναι γενικά δύσκολη υπόθεση. Ας δούμε ένα τμήμα προγράμματος που δίνει μία λύση.

```
..... *
E4 <- E MOD 4
E100 <- E MOD 100
E400 <- E MOD 400
AN ε4=0 TOTE
    δις <- 1
TEΛΟΣ_ΑΝ
AN ε100=0 TOTE
    δις <- 0
TEΛΟΣ_ΑΝ
AN ε400=0 TOTE
    δις <- 1
TEΛΟΣ_ΑΝ
..... * .
```

Όταν το δις είναι 1 το έτος είναι δίσεκτο, όταν είναι 0, δεν είναι.



Στο εργαστήριο

ΔΕ1. Δίνεται ταξινομημένος πίνακας με στοιχεία, 2, 2, 3, 3, 3, 3, 5, 6, 6, 6, 6, 6, 7, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9. Ζητείται να κατασκευαστεί πρόγραμμα θα εμφανίζει την συχνότητα του αριθμού και την θέση του. Σε περίπτωση που ο αριθμός βρίσκεται περισσότερες από μία φορές, η θέση να είναι η πρώτη που παρουσιάζεται.

Για τα δεδομένα που είδη δώσαμε το πρόγραμμά σου θα πρέπει να δίνει τα παρακάτω αποτελέσματα:

Αριθμός	Συχνότητα	Θέση
2	2	1
3	4	3
5	1	7
6	5	8
7	1	13
8	6	14
9	6	20

Υπόδειξη: ακολούθησε τα βήματα του 1ου παραδείγματος του τετραδίου.