ΕΚΠΑΙΔΕΥΤΗΡΙΑ ΓΕΩΡΓΙΟΥ ΖΩΗ
ΛΥΚΕΙΟ

Δομές Δεδομένων

ΤΑΞΗ: Γ΄ ΛΥΚΕΙΟΥ

ΜΑΘΗΜΑ: ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Δεδομένα είναι τα πρώτα στοιχεία που συλλέγουμε χωρίς να έχουν υποστεί οποιαδήποτε επεξεργασία. Για παράδειγμα η μέτρηση και καταγραφή της θερμοκρασίας είναι ένα δεδομένο, το πλήθος των μαθητών μιας τάξης και οι βαθμοί αυτών είναι δεδομένα.

Η επεξεργασία των δεδομένων έχει ως αποτέλεσμα την πληροφορία. Για παράδειγμα, αν πάρουμε τις θερμοκρασίες που μετρήσαμε και βρούμε το μέσο όρο, τότε ο μέσος όρος είναι μια πληροφορία, όπως επίσης είναι πληροφορία και ο μεγαλύτερος βαθμός των μαθητών που προαναφέραμε. Με λίγα λόγια, οι αλγόριθμοι συνήθως παίρνουν κάποια δεδομένα και με επεξεργασία και συσχετισμό αυτών παράγουν πληροφορίες.

Για να μπορούμε να αποθηκεύουμε τα δεδομένα που μαζεύουμε και στη συνέχεια να τα επεξεργαζόμαστε χρησιμοποιούμε τις Δομές Δεδομένων.

Δομή δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που μπορούν και υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

Οι λειτουργίες αυτές είναι:

- Εισαγωγή να βάλουμε νέα δεδομένα
- Διαγραφή να σβήσουμε κάποια δεδομένα
- Προσπέλαση να «δούμε» τα δεδομένα που έχουμε ή αν θέλουμε να τα τροποποιήσουμε. Με πιο απλά λόγια να έχουμε πρόσβαση στα δεδομένα της δομής
- Αντιγραφή να αντιγράψουμε κάποια δεδομένα σε μια άλλη δομή
- Αναζήτηση να βρούμε αν ένα συγκεκριμένο δεδομένο υπάρχει ή όχι μέσα στην δομή που επεξεργαζόμαστε
- Ταξινόμηση να ταξινομήσουμε τα δεδομένα της δομής είτε κατά αύξουσα είτε κατά φθίνουσα σειρά
- Συγχώνευση να ενώσουμε δύο ή περισσότερες δομές σε μία
- Διαχωρισμός να χωρίσουμε μία δομή σε δύο ή περισσότερες νέες δομές.

Οι δομές δεδομένων χωρίζονται σε δύο βασικές κατηγορίες:

- Στατικές δομές δεδομένων οι οποίες έχουν τα παρακάτω χαρακτηριστικά:
  α) έχουν σταθερό μέγεθος κατά τη διάρκεια της εκτέλεσης του προγράμματος.
  β) καταλαμβάνουν συγκεκριμένο χώρο μνήμης.
  γ) τα στοιχεία τους αποθηκεύονται πάντα σε συνεχόμενες θέσεις μνήμης.

- **Δυναμικές δομές δεδομένων**, οι οποίες έχουν τα παρακάτω χαρακτηριστικά:
  α) μπορούν και αλλάζουν (αυξομειώνεται το μέγεθος τους) κατά την εκτέλεση του προγράμματος. Άρα, μπορούμε να εισάγουμε νέα δεδομένα ή να διαγράψουμε παλιά δεδομένα από τη δομή καθώς εκτελείται το πρόγραμμα.
  β) Δεν καταλαμβάνουν συγκεκριμένο χώρο μνήμης, αλλά συνεχώς τους παραχωρείται μνήμη αναλόγως τις ανάγκες τους σε χώρο.
  γ) τα στοιχεία τους δεν αποθηκεύονται αναγκαστικά σε συνεχόμενες θέσεις μνήμης.

Στα πλαίσια του μαθήματος αυτού θα ασχοληθούμε μόνο με στατικές δομές δεδομένων, οι οποίες υλοποιούνται με τη βοήθεια των πινάκων. Οι πίνακες που χρησιμοποιούν είναι ακριβώς όπως τους ξέρουμε από τα μαθηματικά και από την καθημερινή μας ζωή.

Η μόνη διαφορά με την καθημερινή ζωή είναι ότι εδώ οι πίνακες περιέχουν **υποχρεωτικά** στοιχεία που είναι του **ίδιου τύπου**. Για παράδειγμα, μόνο ακέραιους, μόνο πραγματικούς, μόνο λέξεις (αλφαριθμητικός τύπος) κ.λπ.

Τελικά όμως πότε θα χρησιμοποιούμε πίνακες στους αλγόριθμούς μας;

Πίνακες θα πρέπει να χρησιμοποιούμε όταν πραγματικά μας χρειάζεται όπως όταν κάποια δεδομένα θέλουμε να τα επεξεργαστούμε παραπάνω από μία φορά. Για παράδειγμα, έστω ότι θέλουμε να βρούμε το μέσο όρο 50 αριθμών και στη συνέχεια να βρούμε πόσοι αριθμοί είναι μεγαλύτεροι και πόσοι μικρότεροι από το μέσο όρο. Πρέπει πρώτα να διαβάσουμε όλους τους αριθμούς (να τους δώσει ο χρήστης) για να βρούμε το μέσο όρο και στη συνέχεια να τους ξαναδιαβάσουμε (να τους ξαναδώσει ο χρήστης) για να βρούμε πόσοι είναι πάνω και πόσοι κάτω από το μέσο όρο. Σε αυτή την περίπτωση είναι παράλογο να ζητάμε από το χρήστη να δώσει 50 αριθμούς δύο φορές. Γι' αυτό την πρώτη φορά που θα δώσει ο χρήστης τους αριθμούς τους αποθηκεύουμε σε έναν πίνακα, ώστε να είναι διαθέσιμοι στη συνέχεια για περαιτέρω επεξεργασία.

Επίσης πίνακες θα χρησιμοποιούμε όταν αναφέρεται ρητά από την εκφώνηση του προβλήματος.

Οι πίνακες που θα χρησιμοποιήσουμε είναι οι μονοδιάστατοι και οι δισδιάστατοι. Για την καλύτερη κατανόηση των πινάκων καλό θα είναι, όποτε έχουμε έναν πίνακα, να κάνουμε μια πρόχειρη γραφική αναπαράσταση, ώστε να καταλαβαίνουμε πώς «κινείται» ο αλγόριθμος μέσα στον πίνακα.

Ουσιαστικά οι πίνακες είναι μια ομάδα μεταβλητών που έχουν το ίδιο όνομα και χρησιμοποιούμε, και κάποιους δείκτες (ακέραιους αριθμούς), για να δείξουμε ποια από όλες αυτές τις μεταβλητές θέλουμε. Έτσι τα στοιχεία ενός πίνακα θα τα επεξεργαζόμαστε σαν να είναι μεταβλητές. Θα τους δίνουμε αρχική τιμή όπου χρειάζεται, θα εκχωρούμε σ' αυτά νέες τιμές, θα τα χρησιμοποιούμε σε εκφράσεις κ.ο.κ. Αν ένα στοιχείο ενός πίνακα δεν έχει πάρει καμία τιμή, τότε η τιμή του θα είναι απροσδιόριστη, όπως ακριβώς συμβαίνει και με τις μεταβλητές.

Μονοδιάστατοι πίνακες

Οι μονοδιάστατοι πίνακες είναι αυτοί που αποτελούνται από μία γραμμή και πολλές στήλες ή από μία στήλη και πολλές γραμμές, όπως ο πίνακας TEMP που φαίνεται στο διπλανό σχήμα. Για να αναφερθούμε σε ένα στοιχείο του πίνακα χρησιμοποιούμε το όνομα του πίνακα και το δείκτη που δείχνει σε ποια θέση είναι το συγκεκριμένο στοιχείο με τον ακόλουθο τρόπο:

TEMP[1], TEMP[5]

	TEMP
1	23
2	25
3	18
4	17
5	21
6	22
7	28
8	29

Για να αναφερθούμε σε όλα τα στοιχεία ενός πίνακα χρησιμοποιούμε την εντολή Για...

```
Για i από 1 μέχρι 8
    Εμφάνισε TEMP[i]
Τέλος_επανάληψης
```

Η παραπάνω επαναληπτική διαδικασία εμφανίζει όλα τα στοιχεία του πίνακα.

Για να αποθηκεύσουμε τα δεδομένα που δίνει ο χρήστης σε έναν πίνακα χρησιμοποιούμε την εντολή **Διάβασε**, όπως ακριβώς κάνουμε και με τις μεταβλητές.

```
Για i από 1 μέχρι 30
    Εμφάνισε 'Δώσε την ', i, 'η θερμοκρασία'
    Διάβασε ΘΕΡΜ[i]
Τέλος_επανάληψης
```

Η παραπάνω επαναληπτική διαδικασία δημιουργεί έναν πίνακα 30 θέσεων από τα στοιχεία που δίνει ο χρήστης.

Ας δούμε στη συνέχεια μερικές βασικές λειτουργίες που εφαρμόζονται στους μονοδιάστατους πίνακες.

Εύρεση μέγιστου και ελάχιστου στοιχείου του πίνακα ΠΙΝ, Ν θέσεων.

```
Αλγόριθμος ΜέγιστοΕλάχιστο
Δεδομένα //ΠΙΝ, Ν//
max←ΠΙΝ[1]
min←ΠΙΝ[1]
Για i από 1 μέχρι Ν
    Αν ΠΙΝ[i] > max τότε
        max←ΠΙΝ[i]
    Τέλος_αν
    Αν ΠΙΝ[i] < min τότε
        min←ΠΙΝ[i]
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα //max, min//
Τέλος ΜέγιστοΕλάχιστο
```

Στον παραπάνω αλγόριθμο θέτουμε ως αρχική τιμή των μεταβλητών max και min το πρώτο στοιχείο του πίνακα.

Επίσης είναι η πρώτη φορά που βλέπουμε τα δεδομένα και τα αποτελέσματα.

Τα δεδομένα χρησιμοποιούνται όταν θέλουμε να δηλώσουμε ότι ένας αλγόριθμος χρησιμοποιεί κάποια δεδομένα, τα οποία δίνονται από την αρχή του αλγόριθμου. Στον παραπάνω αλγόριθμο στα δεδομένα δηλώνουμε ότι στον αλγόριθμο πρέπει να δώσουμε έναν πίνακα με το όνομα ΠΙΝ και το πλήθος των στοιχείων του πίνακα (Ν).

Τα αποτελέσματα τα χρησιμοποιούμε για να δηλώσουμε ποια είναι τα αποτελέσματα τα αλγόριθμου όταν αυτά δεν είναι εμφανή, δηλαδή δεν εμφανίζονται με μια εντολή **Εμφάνισε.**

Εύρεση του αθροίσματος και του μέσου όρου ενός πίνακα αριθμών.

```
Αλγόριθμος άθροισμα_μέσος
Δεδομένα //ΠΙΝ, Ν//
sum←0
Για i από 1 μέχρι Ν
    sum←sum + ΠΙΝ[i]
Τέλος_επανάληψης
ΜΟ←sum/N
Αποτελέσματα //sum, MO//
Τέλος άθροισμα_μέσος
```

Στους πίνακες πρέπει να είμαστε προσεκτικοί με τα όρια του πίνακα, δηλαδή να προσέχουμε να μην αναφερθούμε σε στοιχείο του πίνακα το οποίο δεν υπάρχει, όπως είναι το στοιχείο στην θέση 0. Επίσης πρέπει να προσέχουμε, όταν αναφερόμαστε στα στοιχεία ενός πίνακα, αυτά να έχουν πάρει μία τιμή.

Δισδιάστατοι πίνακες

Οι δισδιάστατοι πίνακες είναι αυτοί που αποτελούνται από πολλές γραμμές και πολλές στήλες, όπως ο πίνακας ΠΙΝ που φαίνεται στο διπλανό σχήμα.

	1	2	3	4	5	6	7
1	23	25	68	95	7	0	23
2	24	36	2	78	9	45	25
3	36	34	31	25	18	19	24
4	21	22	22	23	28	31	27
5	30	24	31	29	30	32	24

Όταν ο πίνακας έχει ίδιο αριθμό γραμμών και στηλών ονομάζεται τετραγωνικός.

Όλα τα στοιχεία του πίνακα πρέπει οπωσδήποτε να είναι του ίδιου τύπου δεδομένων. Έτσι δεν μπορούμε να έχουμε έναν πίνακα που μία στήλη να περιέχει ονόματα και μία άλλη στήλη να περιέχει αριθμούς.

Για να αναφερθούμε σε ένα στοιχείο του πίνακα χρησιμοποιούμε το όνομα του πίνακα και τους δείκτες που δείχνουν σε ποια θέση είναι το συγκεκριμένο στοιχείο, όπως ακριβώς γίνεται σε μία σκακιέρα όταν θέλουμε να πούμε σε ποια θέση βρίσκεται κάποιο πιόνι.

Στους πίνακες πρώτα αναφέρουμε την γραμμή και μετά τη στήλη στην οποία βρίσκεται το στοιχείο που θέλουμε:

ΠΙΝ[1, 5], ΠΙΝ[4, 5]

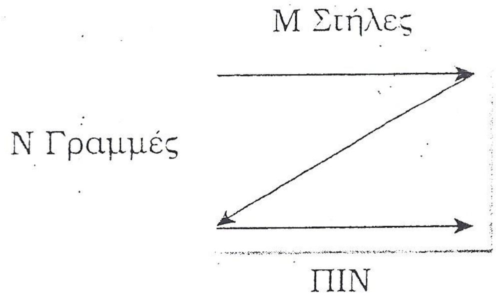
Για να δουλέψουμε με έναν δισδιάστατο πίνακα χρειαζόμαστε δύο εμφωλευμένες επαναληπτικές διαδικασίες, η μία για να καθορίζει την κίνηση μέσα στον πίνακα κατά γραμμές και η άλλη κατά στήλες.

Θα πρέπει κάθε φορά να καταλαβαίνουμε, ποια είναι η ροή του αλγόριθμου μέσα στον πίνακα, πότε γίνεται αλλαγή γραμμής και πότε αλλαγή στήλης, ώστε να μπορούμε να δουλέψουμε μεμονωμένα για κάθε μια στήλη ή γραμμή.

Ας δούμε όμως τις επαναληπτικές εντολές και την κίνηση τους μέσα στον πίνακα, για έναν τυχαίο πίνακα N×M (N γραμμές και M στήλες)

α) Κατά γραμμές

Για i **από** 1 **μέχρι** N
    Για j **από** 1 **μέχρι** M
        Εμφάνισε ΠΙΝ[i,j]
    Τέλος_επανάληψης
Τέλος_Επανάληψης

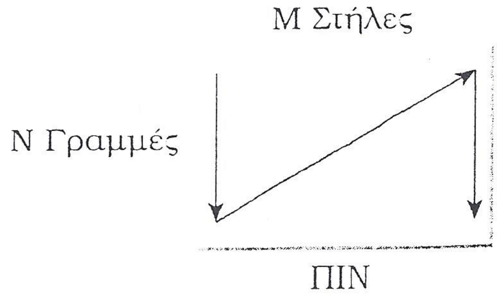


Στις παραπάνω επαναληπτικές διαδικασίες το i αντιστοιχεί στις γραμμές και το j στις στήλες. Αρχικά το i γίνεται ίσο με 1 (πρώτη γραμμή) και αρχίζει να μεταβάλεται το j από 1 μέχρι M (πάει σε όλα τα στοιχεία της πρώτης γραμμής). Όταν τελειώσει το j (έχει τελειώσει η πρώτη γραμμή), αυξάνει το i κατά 1 (πάει στην επόμενη γραμμή) και ξαναρχίζει τα ίδια για το j.

Με βάση αυτά που είπαμε, οι παραπάνω επαναληπτικές διαδικασίες εμφανίζουν κατά γραμμές όλα τα στοιχεία του πίνακα ΠΙΝ.

β) Κατά στήλες

Για j **από** 1 **μέχρι** M
    Για i **από** 1 **μέχρι** N
        Εμφάνισε ΠΙΝ[i,j]
    Τέλος_επανάληψης
Τέλος_Επανάληψης



Σε αυτή την περίπτωση κρατήσαμε το δείκτη i για τις γραμμές και τον δείκτη j για τις στήλες και αλλάξαμε την σειρά των επαναληπτικών διαδικασιών. Η αλλαγή αυτή έχει ως αποτέλεσμα να αλλάζει η ροή μέσα στον πίνακα και να πηγαίνει κατά στήλες.

Έτσι οι παραπάνω επαναληπτικές διαδικασίες εμφανίζουν κατά στήλες όλα τα στοιχεία του πίνακα ΠΙΝ.

! *ΚΑΝΟΝΑΣ: Στα στοιχεία του πίνακα αναφέρουμε πάντοτε πρώτα τις γραμμές και μετά τις στήλες (ΠΙΝ[i,j]). Όταν θέλουμε να κινηθούμε κατά γραμμές, βάζουμε πρώτα την επαναληπτική διαδικασία που αναφέρεται στις γραμμές, ενώ, όταν θέλουμε να κινηθούμε κατά στήλες βάζουμε πρώτα την επαναληπτική διαδικασία που αναφέρεται στις στήλες.*

Βασικές λειτουργίες σε δισδιάστατους πίνακες

Άθροισμα όλων των στοιχείων του πίνακα

Αλγόριθμος άθροισμα
Δεδομένα //ΠΙΝ, Ν, Μ//
sum←0
Για i **από** 1 **μέχρι** Ν
    Για j **από** 1 **μέχρι** Μ
        sum←sum + ΠΙΝ[i,j]
    Τέλος_επανάληψης
Τέλος_Επανάληψης
Εμφάνισε 'Το άθροισμα των στοιχείων του πίνακα είναι: ', sum
Τέλος άθροισμα

Άθροισμα των στοιχείων της κάθε γραμμής του πίνακα

Αλγόριθμος άθροισμα_γραμμής
Δεδομένα //ΠΙΝ, Ν, Μ//
Για i **από** 1 **μέχρι** Ν

```
        sum←0
        Για j από 1 μέχρι M
            sum←sum + ΠΙΝ[i,j]
        Τέλος_επανάληψης
        Εμφάνισε 'Το άθροισμα της ', i, 'ης γραμμής είναι: ', sum
    Τέλος_Επανάληψης
    Τέλος άθροισμα_γραμμής
```

Άθροισμα των στοιχείων της κάθε στήλης του πίνακα

```
Αλγόριθμος άθροισμα_στήλης
Δεδομένα //ΠΙΝ, Ν, Μ//
Για j από 1 μέχρι M
    sum←0
    Για i από 1 μέχρι N
        sum←sum + ΠΙΝ[i,j]
    Τέλος_επανάληψης
    Εμφάνισε 'Το άθροισμα της ', j, 'ης στήλης είναι: ', sum
Τέλος_Επανάληψης
Τέλος άθροισμα_στήλης
```

Μέσος όρος όλων των στοιχείων του πίνακα

```
Αλγόριθμος Μέσος
Δεδομένα //ΠΙΝ, Ν, Μ//
sum←0
Για i από 1 μέχρι N
    Για j από 1 μέχρι M
        sum←sum + ΠΙΝ[i,j]
    Τέλος_επανάληψης
Τέλος_Επανάληψης
ΜΟ←sum/(N*M)
Εμφάνισε 'Ο μέσος όρος των στοιχείων του πίνακα είναι:', ΜΟ
Τέλος Μέσος
```

Μέσος όρος των στοιχείων της κάθε γραμμής του πίνακα

```
Αλγόριθμος Μέσος_γραμμής
Δεδομένα //ΠΙΝ, Ν, Μ//
Για i από 1 μέχρι Ν
    sum←0
    Για j από 1 μέχρι Μ
        sum←sum + ΠΙΝ[i,j]
    Τέλος_επανάληψης
    ΜΟ←sum/Μ
    Εμφάνισε 'Ο μέσος όρος της', i, 'ης γραμμής είναι:', ΜΟ
Τέλος_Επανάληψης
Τέλος Μέσος_γραμμής
```

Μέσος όρος των στοιχείων της κάθε στήλης του πίνακα

```
Αλγόριθμος Μέσος_στήλης
Δεδομένα //ΠΙΝ, Ν, Μ//
Για j από 1 μέχρι Μ
    sum←0
    Για i από 1 μέχρι Ν
        sum←sum + ΠΙΝ[i,j]
    Τέλος_επανάληψης
    ΜΟ←sum/Ν
    Εμφάνισε 'Ο μέσος όρος της', j, 'ης στήλης είναι:', ΜΟ
Τέλος_Επανάληψης
Τέλος Μέσος_στήλης
```

Μέγιστο και ελάχιστο στοιχείο του πίνακα

```
Αλγόριθμος Μέγιστο_Ελάχιστο
Δεδομένα //ΠΙΝ, Ν, Μ//
max←ΠΙΝ[1,1]
min←ΠΙΝ[1,1]
Για i από 1 μέχρι Ν
```

```
        Για j από 1 μέχρι M
            Αν ΠΙΝ[i,j] > max τότε
                max← ΠΙΝ[i,j]
            Τέλος_αν
            Αν ΠΙΝ[i,j] < min τότε
                min← ΠΙΝ[i,j]
            Τέλος_αν
        Τέλος_επανάληψης
    Τέλος_Επανάληψης
    Εμφάνισε 'Το μέγιστο στοιχείο του πίνακα είναι:', max
    Εμφάνισε 'Το ελάχιστο στοιχείο του πίνακα είναι:', min
    Τέλος Μέγιστο_Ελάχιστο
```

Μέγιστο και ελάχιστο στοιχείο της κάθε γραμμής του πίνακα

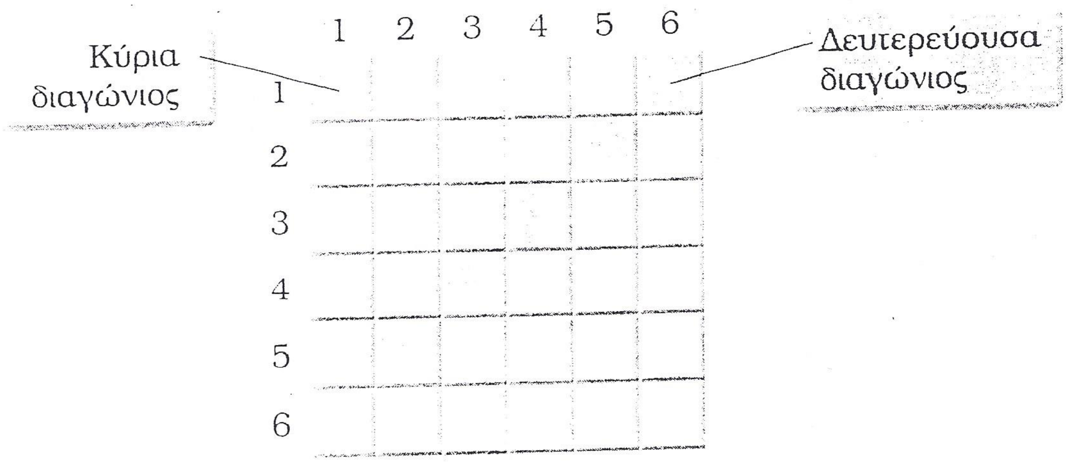
```
Αλγόριθμος Μέγιστο_Ελάχιστο_γραμμής
Δεδομένα //ΠΙΝ, N, M//
Για i από 1 μέχρι N
    max←ΠΙΝ[i,1]
    min←ΠΙΝ[i,1]
    Για j από 1 μέχρι M
        Αν ΠΙΝ[i,j] > max τότε
            max← ΠΙΝ[i,j]
        Τέλος_αν
        Αν ΠΙΝ[i,j] < min τότε
            min← ΠΙΝ[i,j]
        Τέλος_αν
    Τέλος_επανάληψης
    Εμφάνισε 'Το μέγιστο στοιχείο της', i, ' γραμμής είναι:', max
    Εμφάνισε 'Το ελάχιστο στοιχείο της', i, ' γραμμής είναι:', min
Τέλος_Επανάληψης
Τέλος Μέγιστο_Ελάχιστο_γραμμής
```

Μέγιστο και ελάχιστο στοιχείο της κάθε στήλης του πίνακα

```
Αλγόριθμος Μέγιστο_Ελάχιστο_στήλης
Δεδομένα //ΠΙΝ, Ν, Μ//
Για j από 1 μέχρι Μ
    max←ΠΙΝ[1,j]
    min←ΠΙΝ[1,j]
    Για i από 1 μέχρι Ν
        Αν ΠΙΝ[i,j] > max τότε
            max← ΠΙΝ[i,j]
        Τέλος_αν
        Αν ΠΙΝ[i,j] < min τότε
            min← ΠΙΝ[i,j]
        Τέλος_αν
    Τέλος_επανάληψης
    Εμφάνισε 'Το μέγιστο στοιχείο της', j, ' στήλης είναι:', max
    Εμφάνισε 'Το ελάχιστο στοιχείο της', j, ' στήλης είναι:', min
Τέλος_Επανάληψης
Τέλος Μέγιστο_Ελάχιστο_στήλης
```

Τετραγωνικοί πίνακες

Όπως είπαμε οι τετραγωνικοί πίνακες είναι αυτοί που έχουν ίσο αριθμό γραμμών και στηλών. Στους τετραγωνικούς πίνακες δουλεύουμε όπως στους δισδιάστατους, με μοναδική προσθήκη την κύρια και την δευτερεύουσα διαγώνιο του πίνακα, οι οποίες ορίζονται μόνο στους τετραγωνικούς πίνακες.



Αν παρατηρήσουμε τους δείκτες των στοιχείων της κύριας διαγωνίου, ([1,1], [2,2], [3,3], [4,4], [5,5], [6,6]) θα καταλάβουμε ότι ένα στοιχείο ανήκει στην κύρια διαγώνιο, όταν ο δείκτης της γραμμής και της στήλης είναι ο ίδιος.

Το ΠΙΝ[i,j] *ανήκει στην κύρια διαγώνιο άρα* i = j

όπου ΠΙΝ *τετραγωνικός πίνακας* N×N

Αν κάνουμε το ίδιο για τη δευτερεύουσα διαγώνιο που τα στοιχεία της είναι στο παράδειγμά μας τα [1,6], [2,5], [3,4], [4,3], [5,2] και το [6,1], θα καταλάβουμε ότι το άθροισμα των δεικτών των στοιχείων της δευτερεύουσας διαγωνίου είναι κατά 1 μεγαλύτερο από τη διάσταση του πίνακα.

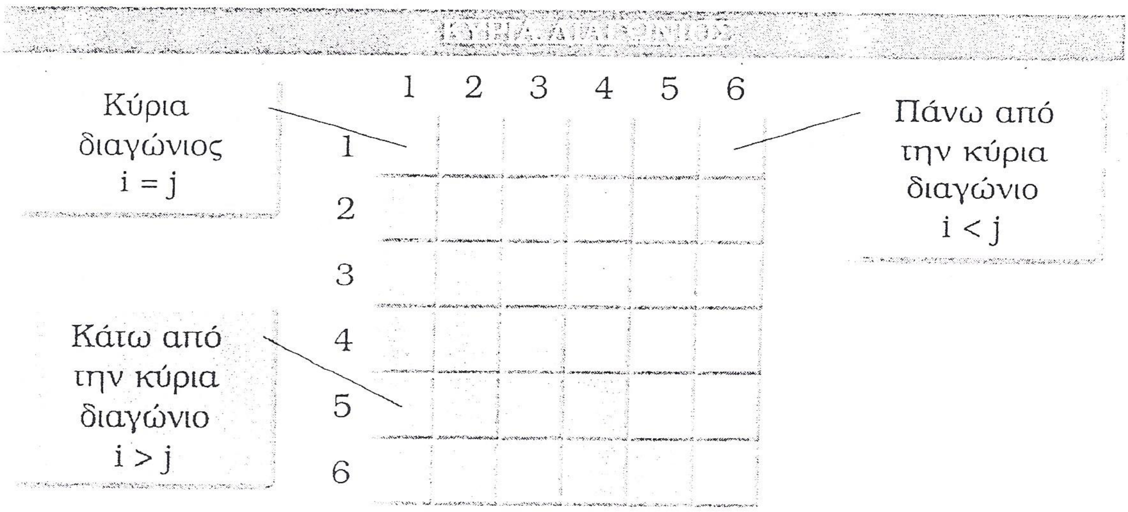
Το ΠΙΝ[i,j] *ανήκει στην δευτερεύουσα διαγώνιο άρα* i + j = N + 1

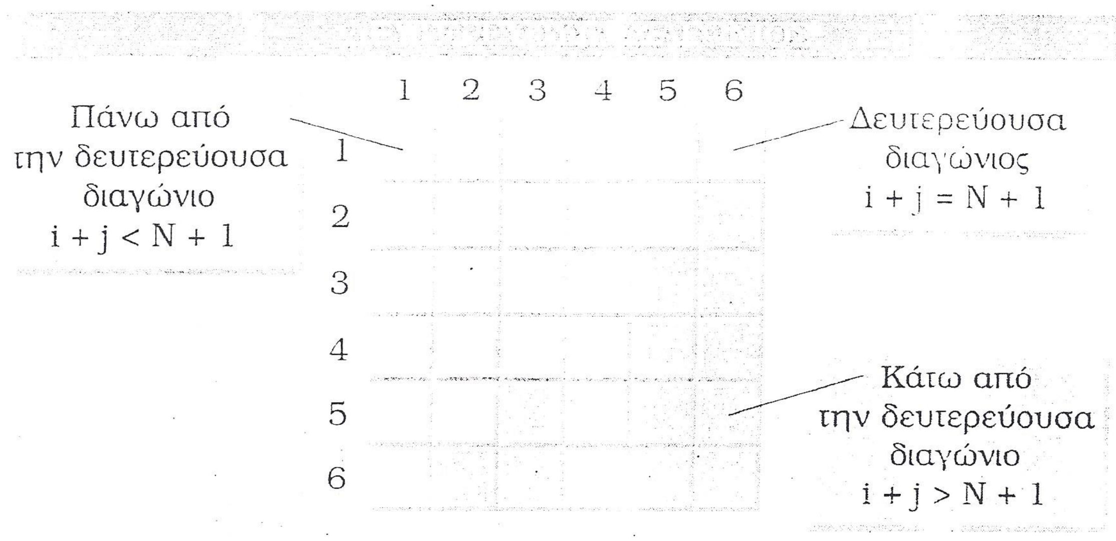
όπου ΠΙΝ τετραγωνικός πίνακας N×N

- Τα στοιχεία της κύριας διαγωνίου μπορούμε να πούμε ότι είναι τα ΠΙΝ[i,i] όπου το i παίρνει τιμές από 1 μέχρι N για τον τετραγωνικό πίνακα ΠΙΝ[N,N]
- Τα στοιχεία της δευτερεύουσας διαγωνίου μπορούμε να πούμε ότι είναι τα ΠΙΝ[i,N+1–i] όπου το i παίρνει τιμές από 1 μέχρι N για τον τετραγωνικό πίνακα ΠΙΝ[N,N]

Με παρόμοια σκέψη μπορούμε να βρούμε την σχέση που συνδέει τους δείκτες των στοιχείων του πίνακα που βρίσκονται πάνω και κάτω από την κύρια και την δευτερεύουσα διαγώνιο του πίνακα.

Οι σχέσεις αυτές φαίνονται στο σχήμα που ακολουθεί





Ταξινόμηση

Η διαδικασία της ταξινόμησης είναι μια από τις πιο συνηθισμένες ενέργειες που εφαρμόζουμε σε έναν πίνακα.

Σκοπός της ταξινόμησης είναι να αλλάξει η σειρά των στοιχείων του πίνακα, ώστε αυτά να εμφανίζονται κατά αύξουσα ή φθίνουσα σειρά.

Το πρόβλημα της ταξινόμησης είναι παλιό, έχει λυθεί και έχουν ήδη διατυπωθεί διάφοροι αλγόριθμοι που επιτυγχάνουν την ταξινόμηση ενός πίνακα.

Στα πλαίσια του μαθήματός μας θα δούμε και θα εξηγήσουμε έναν απλό και διάσημο αλγόριθμο ταξινόμησης, τον αλγόριθμο της φυσσαλίδας (bubble sort). Το όνομά του το έχει πάρει από τον τρόπο με τον οποίο ταξινομεί τον πίνακα και έχει ως εξής:

Έστω ο μονοδιάστατος πίνακας ΠΙΝ, 8 θέσεων, που αρχικά είναι όπως φαίνεται στο σχήμα α.

Για να ταξινομήσουμε τον πίνακα ξεκινάμε από το τελευταίο στοιχείο (ΠΙΝ[8]) και το συγκρίνουμε με το από πάνω του (ΠΙΝ[7]). Αν το κάτω είναι μικρότερο από το πάνω τότε τα αντιμεταθέτουμε.

Μετά πάμε στο επόμενο (ΠΙΝ[7]) και το συγκρίνουμε με το στοιχείο που είναι από πάνω του (ΠΙΝ[6]) και αν χρειάζεται τα αντιμεταθέτουμε. Αυτό συνεχίζεται μέχρι να ελέγξουμε το ΠΙΝ[2] με το ΠΙΝ[1].

Όταν τελειώσει αυτή η διαδικασία, το πιο μικρό στοιχείο θα έχει ανέβει (σαν φυσσαλίδα) στην κορυφή του πίνακα, οπότε έχει ταξινομηθεί (σχήμα β).

Μετά κάνουμε πάλι τα ίδια, ώστε να ταξινομήσουμε και το επόμενο στοιχείο, μόνο που δούλεύουμε για τα στοιχεία 2-8 αφού το πρώτο έχει ταξινομηθεί.

83	3	3	3	3	3	3	3
32	83	8	8	8	8	8	8
3	32	83	18	18	18	18	18
46	8	32	83	19	19	19	19
19	46	18	32	83	23	23	23
23	19	46	19	32	83	32	32
18	23	19	46	23	32	83	46
8	18	23	23	46	46	46	83
Σχήμα α	Σχήμα β	Σχήμα γ	Σχήμα δ	Σχήμα ε	Σχήμα στ	Σχήμα ζ	Σχήμα η

Κάθε φορά που ταξινομούμε ένα στοιχείο, την επόμενη ο αλγόριθμος ελέγχει τα υπόλοιπα που δεν έχουν ταξινομηθεί ακόμη.

Με αυτό το σκεπτικό ο αλγόριθμος χρειάζεται δύο επαναληπτικές διαδικασίες για να υλοποιηθεί. Μία που να καθορίζει μέχρι ποιο στοιχείο να ταξινομεί και μία που να φέρνει το μικρότερο στοιχείο στην κορυφή της προς ταξινόμηση περιοχής του πίνακα.

Ο αλγόριθμος της φυσσαλίδας για τον μονοδιάστατο πίνακα table, N θέσεων έχει ως εξής:

```
Αλγόριθμος Φυσσαλίδα
Δεδομένα //table, N//
Για i από 2 μέχρι N
    Για j από N μέχρι i με_βήμα -1
        Αν table[j] < table[j-1] τότε
            temp ← table[j]
            table[j] ← table[j-1]
            table[j-1] ← temp
        Τέλος_αν
```

```
    Τέλος_επανάληψης
  Τέλος_επανάληψης
  Αποτελέσματα //table//
  Τέλος Φυσσαλίδα
```

Η εξωτερική επανάληψη **Για i από 2 μέχρι N**, είναι αυτή που καθορίζει μέχρι ποιο στοιχείο θα ταξινομεί η εσωτερική επανάληψη που κάνει τις απαραίτητες αντιμεταθέσεις στοιχείων. Ξεκινά από 2 για να μην βγαίνει εκτός των ορίων του πίνακα η εσωτερική επανάληψη. Η τελευταία τιμή που παίρνει το j είναι ίση με την τιμή του i. Επομένως, αν το i ξεκινούσε από 1, τότε το στοιχείο table[j–1] που ελέγχει η εσωτερική επανάληψη, θα ήταν στο τέλος το στοιχείο table[0] που δεν υπάρχει.

Η εσωτερική επανάληψη **Για j από N μέχρι i με_βήμα** –1, είναι αυτή που καθορίζει τους ελέγχους που θα γίνουν. Ξεκινά από το N, δηλαδή από το τέλος του πίνακα, και ανεβαίνει προς τα πάνω μέχρι το i, δηλαδή μέχρι εκεί που δεν έχει ταξινομηθεί ο πίνακας.

Η εντολή επιλογής **Αν** table[j] < table[j–1] **τότε** είναι αυτή που φέρνει το μικρότερο στοιχείο στην κορυφή του πίνακα, σαν να είναι μια φυσσαλίδα στον πάτο ενός ποτηριού που ανεβαίνει σιγά – σιγά στην κορυφή. Αποτέλεσμα αυτού είναι να ταξινομηθεί ο πίνακας κατά **αύξουσα σειρά.**

Παρατήρηση: Αν θέλουμε να ταξινομήσουμε τον πίνακα κατά **φθίνουσα σειρά**, απλά αλλάζουμε τη συνθήκη στην εντολή επιλογής και την κάνουμε **Αν** table[j] > table[j–1] **τότε**. **Όλα τα άλλα παραμένουν τα ίδια.**

Αναζήτηση

Η διαδικασία της αναζήτησης είναι και αυτή μία από τις πιο συνηθισμένες ενέργειες που εφαρμόζουμε στους πίνακες.

Την χρησιμοποιούμε όταν θέλουμε να βρούμε αν υπάρχει ή όχι ένα συγκεκριμένο στοιχείο μέσα σε έναν πίνακα.

Αλγορίθμους που να κάνουν αναζήτηση σε έναν μονοδιάστατο πίνακα θα μάθουμε δύο, τη ΣΕΙΡΙΑΚΗ και τη ΔΥΑΔΙΚΗ αναζήτηση.

Σειριακή αναζήτηση

Το σκεπτικό της σειριακής αναζήτησης είναι απλό. Παίρνει ένα-ένα τα στοιχεία του πίνακα και ελέγχει αν είναι αυτό που ψάχνει. Η διαδικασία της αναζήτησης συνεχίζεται μέχρι να βρεθεί το στοιχείο ή μέχρι να τελειώσει ο πίνακας.

Για να ελεγχθεί αν έχει τελειώσει ο πίνακας, χρησιμοποιούμε την απλή συνθήκη i <= N, όπου i είναι ο δείκτης που χρησιμοποιούμε για την κίνηση μέσα στον πίνακα και N είναι η διάσταση του πίνακα.

Για να ελεγχθεί αν το στοιχείο έχει βρεθεί ή όχι, θα χρησιμοποιήσουμε μια μεταβλητή λογικού τύπου (Boolean).

Οι μεταβλητές λογικού τύπου είναι αυτές που μπορούν να πάρουν μόνο δύο τιμές (ΑΛΗΘΗΣ – ΨΕΥΔΗΣ ή 0-1). Χρησιμοποιούνται για να απεικονίσουν καταστάσεις που έχουν μόνο δύο αποδεκτές τιμές, όπως στην περίπτωση της αναζήτησης που οι δύο καταστάσεις είναι **έχει βρεθεί** και **δεν έχει βρεθεί**.

Κάνουμε μια αντιστοίχιση των τιμών της μεταβλητής με μία κατάσταση, για παράδειγμα η κατάσταση **έχει βρεθεί**, αντιστοιχεί στην τιμή ΑΛΗΘΗΣ και η κατάσταση **δεν έχει βρεθεί**, αντιστοιχεί στην τιμή ΨΕΥΔΗΣ. Έτσι, αν χρησιμοποιήσουμε τη μεταβλητή done που είναι λογικού τύπου για να ελέγχουμε αν δεν έχει βρεθεί το στοιχείο, θα πρέπει να ελέγχουμε αν η μεταβλητή done έχει την τιμή ΨΕΥΔΗΣ.

Τελικά μπορούμε να πούμε ότι η αναζήτηση συνεχίζεται Όσο δεν έχει τελειώσει ο πίνακας (Όσο i <= N) και Όσο δεν έχει βρεθεί το στοιχείο (Όσο done = ΨΕΥΔΗΣ).

Οπότε ο αλγόριθμος της σειριακής αναζήτησης έχει ως εξής:

```
Αλγόριθμος Σειριακή_Αναζήτηση
Δεδομένα //table, key, N//
i ← 1
done ← ΨΕΥΔΗΣ
pos ← 0
Όσο (i <= N) ΚΑΙ (done = ΨΕΥΔΗΣ) επανάλαβε
    Αν table[i] = key τότε
        done ← ΑΛΗΘΗΣ
        pos ← i
    αλλιώς
        i ← i + 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα //done, pos//
Τέλος Σειριακή_Αναζήτηση
```

Στον παραπάνω αλγόριθμο γίνεται αναζήτηση του στοιχείου key στον πίνακα table N θέσεων.

Η μεταβλητή pos χρησιμοποιείται για να αποθηκεύσουμε την θέση στην οποία βρέθηκε το στοιχείο, ώστε να μπορούμε να ξέρουμε πού ακριβώς έχει βρεθεί αυτό μέσα στον πίνακα.

Αργότερα θα δούμε παραδείγματα, στα οποία είναι απαραίτητη η γνώση της θέσης στην οποία έχει βρεθεί το στοιχείο και θα γίνει πιο κατανοητή η χρησιμότητα της μεταβλητής pos.

Δυαδική αναζήτηση (Μέθοδος διαίρει και βασίλευε)

Η δυαδική αναζήτηση στηρίζεται στη μέθοδο «διαίρει και βασίλευε» η οποία με πιο απλά λόγια λέει ότι, για να λύσεις ένα δύσκολο πρόβλημα, το χωρίζεις σε άλλα μικρότερα τα οποία σαφώς και θα είναι πιο εύκολα.

Καταρχάς βασική προϋπόθεση για να μπορούμε να εφαρμόσουμε δυαδική αναζήτηση σε έναν πίνακα είναι ο πίνακας να είναι ταξινομημένος.

Ας υποθέσουμε ότι έχουμε έναν πίνακα ακεραίων 1000 θέσεων, ο οποίος είναι ταξινομημένος κατά αύξουσα σειρά, και ότι ψάχνουμε να βρούμε αν ένας αριθμός βρίσκεται μέσα σ' αυτό τον πίνακα.

Με τη λογική της σειριακής αναζήτησης θα παίρναμε ένα – ένα με την σειρά τα στοιχεία του πίνακα για να δούμε αν βρίσκεται μέσα στον πίνακα ο αριθμός που ψάχνουμε. Αν για παράδειγμα ο αριθμός βρισκόταν στην $100^{η}$ θέση του πίνακα, τότε θα έπρεπε να κάνουμε 100 ελέγχους μέχρι να το βρούμε.

Ενώ με το σκεπτικό της δυαδικής αναζήτησης θα ενεργούσαμε ως εξής:

Πάμε στη μέση του πίνακα (στο $500^{ο}$ στοιχείο) και βλέπουμε αν το στοιχείο που είναι εκεί είναι αυτό που ψάχνουμε. Αν ναι σταματάμε, αν όχι ελέγχουμε το στοιχείο αυτό με αυτό που ψάχνουμε. Αν το στοιχείο είναι μεγαλύτερο απ' αυτό που ψάχνουμε, τότε αυτό που ψάχνουμε δεν μπορεί να βρίσκεται από τη μέση και κάτω. Έτσι συνεχίζουμε να ψάχνουμε από το $1^{ο}$ μέχρι το $499^{ο}$ στοιχείο του πίνακα και αυτό έχει ως αποτέλεσμα με έναν μόνο έλεγχο να απορρίψουμε 501 θέσεις.

Με παρόμοιο τρόπο ενεργούμε αν το στοιχείο που είναι στη μέση είναι μικρότερο απ' αυτό που ψάχνουμε.

Όσο μεγαλύτερο είναι το μέγεθος του πίνακα, τόσο πιο γρήγορη, κατά μέσο όρο, είναι η δυαδική από τη σειριακή αναζήτηση.

Πότε όμως θα εφαρμόζουμε τη δυαδική αναζήτηση και πότε τη σειριακή:

- Αν ο πίνακας δεν είναι ταξινομημένος, εφαρμόζουμε πάντα τη σειριακή (ΔΕΝ μπορούμε να εφαρμόσουμε τη δυαδική).
- Αν ο πίνακας είναι ταξινομημένος αλλά είναι μικρός (το πολύ 20 θέσεων), επιλέγουμε όποια θέλουμε.
- Αν ο πίνακας είναι ταξινομημένος και μεγάλος, είναι καλύτερα να χρησιμοποιούμε τη δυαδική (μπορούμε να χρησιμοποιούμε και τη σειριακή, αλλά θα καθυστερεί κατά την εκτέλεση ο αλγόριθμος).

Ο αλγόριθμος της δυαδικής αναζήτησης έχει ως εξής:

```
Αλγόριθμος Δυαδική_Αναζήτηση
Δεδομένα //ΠΙΝ, Ν, key//
αρχή ← 1
τέλος ← Ν
done ← ΨΕΥΔΗΣ
pos ← 0
Όσο (done = ΨΕΥΔΗΣ) ΚΑΙ (αρχή <= τέλος) επανάλαβε
    μέσο ← (τέλος + αρχή) DIV 2
    Αν ΠΙΝ[μέσο] = key τότε
        done ← ΑΛΗΘΗΣ
        pos ← μέσο
    αλλιώς_αν ΠΙΝ[μέσο] > key τότε
        τέλος ← μέσο – 1
    αλλιώς
        αρχή ← μέσο + 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα //done, pos//
Τέλος Δυαδική_Αναζήτηση
```

Οι μεταβλητές done και pos έχουν την ίδια χρησιμότητα και λειτουργία με τη σειριακή αναζήτηση.

Μέσα στην επανάληψη βρίσκουμε το μέσο του πίνακα που έχουμε και μετά ελέγχουμε αν στο μέσο είναι αυτό που ψάχνουμε, οπότε και κάνουμε το done, ΑΛΗΘΗ. Αν δεν είναι στο μέσο και το ΠΙΝ[μέσο] είναι μεγαλύτερο από το key, τότε αλλάζουμε τα όρια του πίνακα και κάνουμε το τέλος ίσο με το (μέσο – 1), ώστε στην επόμενη επανάληψη να ψάξει μόνο στο μισό πίνακα που μπορεί να είναι το key. Αν το ΠΙΝ[μέσο] είναι μικρότερο από το key, τότε το key μπορεί να βρίσκεται από τη μέση και κάτω οπότε κάνουμε την αρχή ίση με (μέσο + 1).

Όλα αυτά γίνονται όσο δεν έχουμε βρει το key (Όσο done = ΨΕΥΔΗΣ) ΚΑΙ όσο δεν έχει τελειώσει ο πίνακας (Όσο αρχή <= τέλος).

Παράλληλοι πίνακες

Όπως έχουμε πει, σε έναν πίνακα υπάρχουν στοιχεία που είναι υποχρεωτικά του ίδιου τύπου δεδομένων.

Τι γίνεται όμως στην περίπτωση που θέλουμε να αποθηκεύσουμε διαφορετικά δεδομένα όπως για παράδειγμα, όταν θέλουμε να αποθηκεύσουμε ονόματα και αριθμούς τηλεφώνων;

Στην περίπτωση αυτή χρησιμοποιούμε διαφορετικούς πίνακες, στους οποίους σε κάθε έναν αποθηκεύουμε και έναν τύπο δεδομένων.

Όταν θέλουμε να αποθηκεύσουμε ονόματα και αριθμούς τηλεφώνων χρησιμοποιούμε έναν πίνακα για τα ονόματα και έναν πίνακα για τους αριθμούς τηλεφώνων.

Οι πίνακες αυτοί δεν είναι ανεξάρτητοι μεταξύ τους, άλλα είναι φτιαγμένοι έτσι ώστε στην ίδια θέση που βρίσκεται το όνομα στον πίνακα με τα ονόματα, να βρίσκεται και ο αριθμός τηλεφώνου στον πίνακα με τους αριθμούς.

Δύο ή περισσότεροι τέτοιοι πίνακες ονομάζονται παράλληλοι.

__Παράλληλοι__ ονομάζονται δύο ή περισσότεροι πίνακες, όταν σ' αυτούς έχουμε αποθηκεύσει τα χαρακτηριστικά οντοτήτων με τέτοιο τρόπο, ώστε τα δεδομένα κάθε οντότητας να βρίσκονται σε στοιχεία με την ίδια τιμή δείκτη.

Ας φανταστούμε για παράδειγμα ότι ο Χρυσός Οδηγός του ΟΤΕ, στην ηλεκτρονική του μορφή, χρησιμοποιεί έναν πίνακα για το όνομα του συνδρομητή, έναν για τον αριθμό τηλεφώνου, έναν για την διεύθυνση και έναν για τον ταχυδρομικό κώδικα.

Σε αυτή την περίπτωση οι διαφορετικές οντότητες είναι οι συνδρομητές του ΟΤΕ. Μία οντότητα είστε και εσείς, επομένως αν το όνομά σας βρίσκεται στην 8η θέση του πίνακα με τα ονόματα, τότε στην 8η θέση του κάθε πίνακα υπάρχουν τα δεδομένα που αφορούν εσάς (αριθμός τηλεφώνου, διεύθυνση και ταχυδρομικός κώδικας).

! ΣΥΜΒΟΥΛΗ: *Σε ασκήσεις με πίνακες πάντοτε να κάνετε έναν πρόχειρο πίνακα είτε με υποθετικά στοιχεία είτε χωρίς, ώστε να μπορείτε να κατανοείτε πιο εύκολα τη μορφή του πίνακα και τις ενέργειες που θέλετε να κάνετε σ' αυτόν.*

Π α ρ ά δ ε ι γ μ α 1 (Βασικές διεργασίες σε μονοδιάστατο πίνακα)

Σε ένα θερμοκήπιο ο υπάλληλος καταγράφει τις θερμοκρασίες του χώρου ανά μία ώρα.

Να γίνει αλγόριθμος ο οποίος:

α) Να διαβάζει τις θερμοκρασίες για μία ημέρα.

β) Να υπολογίζει τη μέγιστη και την ελάχιστη θερμοκρασία της ημέρας.

γ) Να υπολογίζει τη μέση θερμοκρασία της ημέρας.

δ) Να υπολογίζει πόσες θερμοκρασίες είναι μεγαλύτερες και πόσες μικρότερες από τη μέση θερμοκρασία.

Λύση

Επειδή τις τιμές που θα δώσει ο χρήστης θα πρέπει να τις επεξεργαστούμε πάνω από μία φορά, για να μη χρειαστεί να τις ξαναδώσει, θα τις αποθηκεύσουμε σε έναν πίνακα. Αν δεν υπήρχε το δ υποερώτημα, η χρήση πίνακα δεν θα ήταν αναγκαία.

Επομένως θα δημιουργήσουμε τον πίνακα ΘΕΡΜ, ο οποίος θα είναι μονοδιάστατος 24 θέσεων, αφού 24 είναι οι μετρήσεις κατά τη διάρκεια της ημέρας. Στην συνέχεια θα βρούμε το μέγιστο, το ελάχιστο και το μέσο όρο του πίνακα. Κατόπιν θα βρούμε πόσα στοιχεία του πίνακα είναι μεγαλύτερα και πόσα μικρότερα από τον μέσο όρο.

	ΘΕΡΜ
1	
2	
3	
.	.
.	.
.	.
22	
23	
24	

Αλγόριθμος Θερμοκήπιο

Για i **από** 1 **μέχρι** 24

  Εμφάνισε 'Δώστε την', i, 'θερμοκρασία'

  Διάβασε ΘΕΡΜ[i]

Τέλος_επανάληψης

```
max ← ΘΕΡΜ[1]
min ← ΘΕΡΜ[1]
Για i από 1 μέχρι 24
    Αν ΘΕΡΜ[i] > max τότε
        max ← ΘΕΡΜ[i]
    Τέλος_αν
    Αν ΘΕΡΜ[i] < min τότε
        min ← ΘΕΡΜ[i]
    Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε 'Η μέγιστη θερμοκρασία της ημέρας είναι:', max
Εμφάνισε 'Η ελάχιστη θερμοκρασία της ημέρας είναι:', min

sum ← 0
Για i από 1 μέχρι 24
    sum ← sum + ΘΕΡΜ[i]
Τέλος_επανάληψης
ΜΟ ← sum/24
Εμφάνισε 'Η μέση θερμοκρασία της ημέρας είναι:', ΜΟ

πάνω ← 0
κάτω ← 0
Για i από 1 μέχρι 24
    Αν ΘΕΡΜ[i] > ΜΟ τότε
        πάνω ← πάνω + 1
    αλλιώς_αν ΘΕΡΜ[i] < ΜΟ τότε
        κάτω ← κάτω + 1
    Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε 'Πάνω από την μέση θερμοκρασία είναι:', πάνω, 'τιμές'
Εμφάνισε 'Κάτω από την μέση θερμοκρασία είναι:', κάτω, 'τιμές'
Τέλος Θερμοκήπιο
```

Π α ρ ά δ ε ι γ μ α 2 (Δημιουργία πινάκων)

Να γίνει αλγόριθμος ο οποίος να δημιουργεί τους παρακάτω πίνακες (ένας για τον κάθε πίνακα).

	ΑΛΦΑ
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64

	ΒΗΤΑ
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32

ΓΑΜΑ

	1	2	3	4	5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

ΔΕΛΤΑ

	1	2	3	4	5
1	1	0	0	0	0
2	2	1	0	0	0
3	2	2	1	0	0
4	2	2	2	1	0
5	2	2	2	2	1

Λύση

Σε κάθε έναν από τους πίνακες αυτούς θα πρέπει να βρούμε την σχέση βάσει της οποίας εισάγονται τα στοιχεία στον πίνακα.

ΑΛΦΑ: Ο πίνακας ΑΛΦΑ είναι μονοδιάστατος, 8 θέσεων, ο οποίος, αν παρατηρήσουμε, θα δούμε ότι σε κάθε θέση του το στοιχείο που περιέχει είναι το τετράγωνο του δείκτη της θέσης αυτής.

Αλγόριθμος ΑΛΦΑ
Για i **από** 1 **μέχρι** 8
    ΑΛΦΑ[i] ← i*i
Τέλος_επανάληψης
Τέλος ΑΛΦΑ

ΒΗΤΑ: Ο ΒΗΤΑ είναι και αυτός μονοδιάστατος, 8 θέσεων. Αν παρατηρήσουμε τα στοιχεία που περιέχει θα δούμε ότι αυτά είναι το τετραπλάσιο του δείκτη της θέσης αυτής.

```
Αλγόριθμος ΒΗΤΑ
Για i από 1 μέχρι 8
    ΒΗΤΑ[i] ← 4*i
Τέλος_επανάληψης
Τέλος ΒΗΤΑ
```

ΓΑΜΑ: Ο πίνακας αυτός είναι ένας τετραγωνικός πίνακας, του οποίου όλα τα στοιχεία είναι μηδέν, εκτός από τα στοιχεία της κύριας διαγωνίου τα οποία είναι ίσα με 1.

```
Αλγόριθμος ΓΑΜΑ
Για i από 1 μέχρι 5
    Για j από 1 μέχρι 5
        Αν i = j τότε
            ΓΑΜΑ[i,j] ← 1
        αλλιώς
            ΓΑΜΑ[i,j] ← 0
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος ΓΑΜΑ
```

ΔΕΛΤΑ: Ο πίνακας αυτός είναι ένας τετραγωνικός πίνακας, του οποίου τα στοιχεία της κύριας διαγωνίου είναι ίσα με 1, αυτά πού είναι πάνω από την κύρια διαγώνιο είναι ίσα με 0 και αυτά που είναι κάτω από την κύρια διαγώνιο είναι ίσα με 2.

```
Αλγόριθμος ΔΕΛΤΑ
Για i από 1 μέχρι 5
    Για j από 1 μέχρι 5
        Αν i = j τότε
            ΔΕΛΤΑ[i,j] ← 1
        αλλιώς_αν i < j τότε
            ΔΕΛΤΑ[i,j] ← 0
```

```
        αλλιώς
            ΔΕΛΤΑ[i,j] ← 2
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος ΔΕΛΤΑ
```

Παράδειγμα 3 (Εύρεση μεγίστου σε παράλληλους πίνακες)

Σε κάποιους αγώνες ακοντισμού, οι καλύτερες επιδόσεις των 8 αθλητών που παίρνουν μέρος καταχωρούνται στον πίνακα ΕΠΙΔ. Στον πίνακα NAME είναι καταχωρημένα τα ονόματα των αθλητών. Οι πίνακες αυτοί είναι φτιαγμένοι έτσι ώστε η επίδοση και το όνομα ενός αθλητή να βρίσκονται σε θέσεις με τον ίδιο δείκτη στους δύο πίνακες.

Να γίνει αλγόριθμος που θα εμφανίζει το όνομα του αθλητή που παίρνει το χρυσό μετάλιο.

Λύση

Οι δύο πίνακες που χρησιμοποιούνται στον συγκεκριμένο αλγόριθμο είναι παράλληλοι. Αυτό που πρέπει να κάνουμε είναι να βρούμε την μέγιστη επίδοση και σε ποια θέση βρίσκεται. Άρα, θα βρούμε το μέγιστο στοιχείο του πίνακα ΕΠΙΔ και σε ποια θέση βρίσκεται, ώστε να εμφανίσουμε το όνομα του αθλητή που βρίσκεται στην ίδια θέση στον πίνακα NAME.

	ΕΠΙΔ		NAME
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
8		8	

```
Αλγόριθμος Ακοντισμός
Δεδομένα //ΕΠΙΔ, NAME//
max ← ΕΠΙΔ[1]
θέσηmax ← 1
Για i από 1 μέχρι 8
    Αν ΕΠΙΔ[i] > max τότε
        max ← ΕΠΙΔ[i]
        θέσηmax ← i
    Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε 'Το χρυσό μετάλλιο παίρνει ο', NAME[θέσηmax]
Τέλος Ακοντισμός
```

Παρατήρηση: Η αρχική τιμή της μεταβλητής θέσηmax είναι υποχρεωτική γιατί, αν ο νικητής ήταν ο πρώτος αθλητής, τότε δεν θα έπαιρνε ποτέ καμία τιμή και έτσι ο αλγόριθμος σε αυτή την περίπτωση δεν θα δούλευε σωστά.

Παράδειγμα 4 (Αναζήτηση σε παράλληλους πίνακες)

Σε μια εταιρεία που απασχολεί 35 υπαλλήλους, για τη διευκόλυνση του ταμία έχουνε φτιάξει δύο πίνακες, τον πίνακα ΟΝΟΜΑ και τον πίνακα ΜΙΣΘΟΣ, οι οποίοι στις ίδιες τιμές δείκτη περιέχουν το όνομα και το μισθό αντίστοιχα ενός υπαλλήλου. Έτσι την ημέρα των πληρωμών, ο ταμίας πληκτρολογεί το όνομα του υπαλλήλου και του εμφανίζεται ο μισθός που παίρνει ο συγκεκριμένος υπάλληλος.

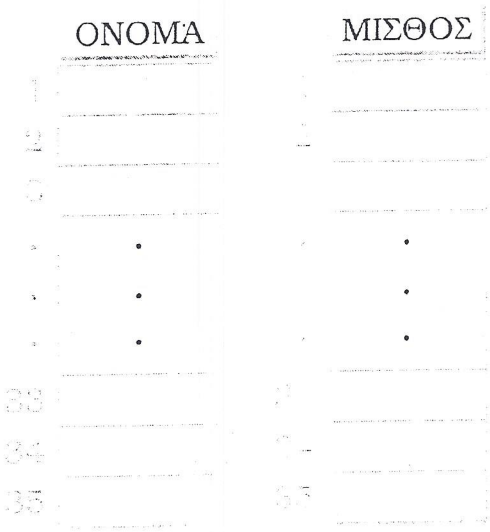
Να αναπτύξετε έναν αλγόριθμο με τον οποίο να μπορεί να δουλεύει το πρόγραμμα.

Λύση

Στον συγκεκριμένο αλγόριθμο θέλουμε να κάνουμε αναζήτηση σε παράλληλους πίνακες.

Θα κάνουμε αναζήτηση του ονόματος στον πίνακα ΟΝΟΜΑ και θα εμφανίσουμε το μισθό που βρίσκεται στον πίνακα ΜΙΣΘΟΣ στην ίδια θέση με αυτήν που βρήκαμε το όνομα.

Άρα πρόκειται για έναν αλγόριθμο με απλή εφαρμογή της σειριακής αναζήτησης.



Αλγόριθμος Ταμείο
Δεδομένα //ΟΝΟΜΑ, ΜΙΣΘΟΣ//
Εμφάνισε 'Δώστε το όνομα του υπαλλήλου'
Διάβασε name
i ← 1
done ← ΨΕΥΔΗΣ
pos ← 0
Όσο (i <= 35) **ΚΑΙ** (done = ΨΕΥΔΗΣ) **επανάλαβε**
    Αν ΟΝΟΜΑ[i] = name **τότε**

```
            done ← ΑΛΗΘΗΣ
            pos ← i
        αλλιώς
            i ← i + 1
        Τέλος_αν
    Τέλος_επανάληψης
    Αν done = ΨΕΥΔΗΣ τότε
        Εμφάνισε 'Δεν υπάρχει υπάλληλος με τέτοιο όνομα'
    αλλιώς
        Εμφάνισε 'Ο μισθός του υπαλλήλου είναι:', ΜΙΣΘΟΣ[pos]
    Τέλος_αν
    Τέλος Ταμείο
```

Παράδειγμα 5 (Ταξινόμηση σε παράλληλους πίνακες)

Στους σχολικούς αγώνες μιας περιφέρειας λαμβάνουν μέρος στους τελικούς των 100 μέτρων 16 παιδιά. Τα ονόματα των παιδιών είναι καταχωρημένα στον πίνακα ΝΑΜΕ. Για την εύκολη επεξεργασία των αποτελεσμάτων οι διοργανωτές έφτιαξαν και τον πίνακα ΕΠΙΔ που περιέχει τις επιδόσεις των 16 παιδιών. Οι πίνακες ΝΑΜΕ και ΕΠΙΔ είναι φτιαγμένοι έτσι ώστε το όνομα ενός παιδιού και η επίδοσή του να βρίσκονται σε θέσεις με τους ίδιους δείκτες στους δύο πίνακες.

Να αναπτύξετε έναν αλγόριθμο ο οποίος να εμφανίζει το όνομα του παιδιού που παίρνει το χάλκινο μετάλλιο.

Λύση

Τα βασικά στοιχεία που πρέπει να προσέξουμε σε αυτό τον αλγόριθμο είναι:

1. Επειδή στο αγώνισμα των 100 μέτρων όσο μικρότερη είναι η επίδοση τόσο πιο καλή είναι, θα ψάξουμε να βρούμε την τρίτη μικρότερη επίδοση στον πίνακα ΕΠΙΔ.
2. Η εύρεση της μικρότερης επίδοσης είναι εύκολη. Όταν όμως θέλουμε να βρούμε την τρίτη μικρότερη, θα πρέπει να ταξινομήσουμε τον πίνακα ΕΠΙΔ κατά αύξουσα σειρά, ώστε η τρίτη καλύτερη επίδοση να βρεθεί στην τρίτη θέση του πίνακα.
3. Ο πίνακας ΕΠΙΔ που θέλουμε να ταξινομήσουμε είναι παράλληλος πίνακας με τον πίνακα ΝΑΜΕ. Επομένως, για να μην μπερδέψουμε την αντιστοίχιση που υπάρχει μεταξύ των δύο πινάκων, θα πρέπει κάθε φο-

ρά που κάνουμε μια αλλαγή στον πίνακα ΕΠΙΔ που θέλουμε να ταξινομήσουμε, την ίδια αλλαγή να την κάνουμε και στον πίνακα NAME. Έτσι, για να εμφανίσουμε το όνομα του παιδιού που παίρνει το χάλκινο μετάλλιο, αρκεί να εμφανίσουμε το όνομα που θα βρίσκεται στην τρίτη θέση του πίνακα NAME, μετά την ταξινόμηση.

Συνεπώς, με βάση τα παραπάνω, ο αλγόριθμος είναι εφαρμογή της φυσαλίδας σε παράλληλους πίνακες και έχει ως εξής:

	NAME		ΕΠΙΔ
1		1	
2		2	
3		3	
.		.	
.		.	
.		.	
14		14	
15		15	
16		16	

Αλγόριθμος Αγώνες
Δεδομένα //ΕΠΙΔ, NAME//
Για i **από** 2 **μέχρι** 16
    Για j **από** 16 **μέχρι** i **με_βήμα** –1
        Αν ΕΠΙΔ[j] < ΕΠΙΔ[j–1] **τότε**
            temp1 ← ΕΠΙΔ[j]
            ΕΠΙΔ[j] ← ΕΠΙΔ[j–1]
            ΕΠΙΔ[j–1] ← temp1
            temp2 ← NAME[j]
            NAME[j] ← NAME[j–1]
            NAME[j–1] ← temp2
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Εμφάνισε 'Το χάλκινο μετάλλιο παίρνει ο:', NAME[3]
Τέλος Αγώνες

Άλυτα προβλήματα

Ασκήσεις

1 Έστω πίνακας table με m γραμμές και n στήλες. Να γραφεί αλγόριθμος ο οποίος να υπολογίζει το μέγιστο στοιχείο του κατά γραμμή και κατά στήλη.

2 Ένας καθηγητής πληροφορικής, έχοντας ως σκοπό να βγάλει κάποια συμπεράσματα από τους βαθμούς των μαθητών του στις πανελλήνιες εξετάσεις, καταχώρησε τους βαθμούς των 35 μαθητών του σε έναν μονοδιάστατο πίνακα με το όνομα ΒΑΘ. Για να βγάλει τα συμπεράσματα που θέλει, πρέπει να υπολογίσει τον καλύτερο βαθμό, τον χειρότερο, τον μέσο όρο καθώς και το εύρος των βαθμών.
Να αναπτύξετε αλγόριθμο που θα υπολογίζει τα παραπάνω.

3 Σε ένα Fast Food καταχωρούνται οι παραγγελίες στον υπολογιστή με βάση τον κωδικό των φαγητών. Συγκεκριμένα, χρησιμοποιείται ένας αλγόριθμος ο οποίος λειτουργεί ως εξής: Διαβάζει τον κωδικό του φαγητού (1-20) και την ποσότητα και εμφανίζει το όνομα του φαγητού και την τιμή του. Η προηγούμενη διαδικασία επαναλαμβάνεται μέχρι να δοθεί ως κωδικός η τιμή 0. Στο τέλος ο αλγόριθμος εμφανίζει τη συνολική αξία της παραγγελίας. Να γραφεί αλγόριθμος που να πραγματοποιεί τις παραπάνω λειτουργίες. Να θεωρήσετε δεδομένους τους πίνακες *Φαγητό* και *Τιμή*, οι οποίοι περιέχουν τα ονόματα και τις τιμές, αντίστοιχα, των 20 φαγητών.

4 Στον κεντρικό υπολογιστή του τμήματος μηχανογράφησης ενός ασφαλιστικού ταμείου καταχωρούνται οι ασφαλισμένοι με όλα τα απαραίτητα στοιχεία τους. Για το σκοπό αυτό χρησιμοποιούνται διαφορετικοί πίνακες, μεταξύ των οποίων, ο πίνακας *Όνομα* και ο πίνακας *Έτος*, οι οποίοι περιέχουν για κάθε ασφαλισμένο το όνομα και το έτος γέννησής του, αντίστοιχα. Γνωρίζοντας ότι κάποιος ασφαλισμένος συνταξιοδοτείται μόλις συμπληρώσει το 65ο έτος της ηλικίας του, να γράψετε αλ-

γόριθμο που:

α) να διαβάζει το τρέχον έτος,

β) να διαβάζει τα ονόματα και τις χρονολογίες γέννησης 100 ασφαλισμένων, και

γ) να εμφανίζει τα ονόματα και το πλήθος εκείνων που θα συνταξιοδοτηθούν σε λιγότερο από 10 χρόνια.

Σημείωση: Οι ασφαλισμένοι που έχουν ήδη συνταξιοδοτηθεί να μην συμπεριληφθούν καθόλου στο υποερώτημα γ).

5 Να γραφεί παραλλαγή για τον αλγόριθμο της σειριακής αναζήτησης, έτσι ώστε να διασχίζεται ολόκληρος ο πίνακας, να εμφανίζονται οι θέσεις στις οποίες υπάρχει ο αριθμός που αναζητείται και να υπολογίζεται η συχνότητα εμφάνισής του.

6 Να επεκταθεί ο αλγόριθμος της σειριακής αναζήτησης σε δισδιάστατο πίνακα n×m.

7 Δίνεται μονοδιάστατος πίνακας Ν στοιχείων. Να γραφεί αλγόριθμος που να τον αντιμεταθέτει.

8 Έστω Α μονοδιάστατος πίνακας ακεραίων με Ν στοιχεία. Να γραφεί αλγόριθμος, ο οποίος να κατασκευάζει έναν δεύτερο πίνακα Β που να περιέχει τα στοιχεία του πίνακα Α με την ίδια σειρά, έχοντας όμως τα μηδενικά μαζεμένα στο τέλος του. Π.χ. αν ο πίνακας Α είναι της μορφής:

A = [1 0 3 7 0 0 6 4 0 9]

τότε ο πίνακας Β θα πρέπει να είναι της μορφής:

B = [1 3 7 6 4 9 0 0 0 0]

9 Έστω μονοδιάστατος πίνακας A[35] ο οποίος περιέχει μια φράση, όπως για παράδειγμα:

A = Α Ν Α Π Τ Υ Ξ Η Ε Φ Α Ρ Μ Ο Γ Ω Ν .

Κάθε στοιχείο του πίνακα περιέχει ένα χαρακτήρα, ενώ το τέλος της φράσης δίνεται με τον χαρακτήρα '.' (τελεία). Να γραφεί αλγόριθμος που να διαβάζει μια φράση (αγνοώντας τα κενά) και να αναγνωρίζει αν η συγκεκριμένη φράση αποτελεί καρκινική επιγραφή.

10 Σε κάποιο πρωτάθλημα άρσης βαρών συμμετέχουν 12 αθλητές. Στο τέλος βραβεύονται οι τρεις καλύτεροι αθλητές στο σύνολο. Σε περίπτωση που δύο ή περισσότεροι αθλητές έχουν την ίδια επίδοση, παίρνει καλύτερη θέση αυτός με το μικρότερο βάρος. Να γραφεί αλγόριθμος ο οποίος:

α) διαβάζει τα ονόματα των 12 αθλητών, το βάρος του καθενός, καθώς και την επίδοσή του στο σύνολο,

β) εμφανίζει τα ονόματα και τις επιδόσεις των αθλητών που παίρνουν μετάλλιο.

11 Οι «τιμές κλεισίματος» μιας μετοχής κατά τη διάρκεια ενός μήνα καταχωρούνται στο μονοδιάστατο πίνακα ΑΞΙΑ. Να γραφεί αλγόριθμος που να υπολογίζει τη μέγιστη άνοδο και τη μέγιστη πτώση, επί τοις εκατό, της συγκεκριμένης μετοχής μεταξύ δύο διαδοχικών ημερών, μέσα στο συγκεκριμένο μήνα. Να υποθέσετε ότι μέσα σ' ένα μήνα το χρηματιστήριο συνεδριάζει 25 φορές.

12 Θεωρήστε τους πίνακες ΜΕΤΟΧΗ και ΑΞΙΑ, οι οποίοι περιέχουν τα ονόματα 150 μετοχών και τις «τιμές κλεισίματός» τους για έναν μήνα, αντίστοιχα. Να γράψετε αλγόριθμο που να εμφανίζει το όνομα της κάθε μετοχής μαζί με τη μέγιστη τιμή που είχε μέσα στο μήνα. Να θεωρήσετε ότι μέσα σ' ένα μήνα το χρηματιστήριο συνεδριάζει 25 φορές.

13 Ρίχνουμε ένα ζάρι Ν φορές και σημειώνουμε τις ενδείξεις. Να γράψετε αλγόριθμο που:

α) Να διαβάζει το πλήθος των ρίψεων.

β) Να διαβάζει τις ενδείξεις, ελέγχοντας αν είναι στο διάστημα [1,6], εμφανίζοντας σχετικό μήνυμα λάθους.

γ) Να μετράει πόσες φορές ήρθε η κάθε ένδειξη και να εμφανίζει τα αποτελέσματα.

14 Στους αγώνες του Golden League στίβου γίνονται 6 αγώνες και αν κάποιος κερδίσει 4 χρυσά μετάλλια, τότε συμμετέχει στο επιπλέον έπαθλο που είναι κάποιοι ράβδοι χρυσού. Στο αγώνισμα της σφαιροβολίας οι επιδόσεις των 15 συνολικά αθλητών που έχουν συμμετάσχει σε

τουλάχιστον 1 τελικό, είναι καταχωρημένες στον πίνακα *ΕΠΙΔ* που περιέχει τις επιδόσεις των αθλητών και στους 6 τελικούς (αν κάποιος αθλητής δεν συμμετείχε σε έναν τελικό τότε ως επίδοση καταχωρείται το 0). Τα ονόματα των αθλητών καταχωρούνται στον πίνακα *NAME* που είναι φτιαγμένος έτσι ώστε, αν το όνομα του αθλητή είναι στην 3η θέση του πίνακα *NAME*, οι επιδόσεις του αθλητή να βρίσκονται στην 3η γραμμή του πίνακα *ΕΠΙΔ*.

Να αναπτύξετε αλγόριθμο ο οποίος να ελέγχει, αν κάποιος από τους αθλητές της σφαιροβολίας θα πάρει το επιπλέον έπαθλο και αν ναι, να εμφανίζει το όνομά του, αλλιώς να εμφανίζει το μήνυμα 'Δεν υπάρχει αθλητής με 4 νίκες'.

15 Ένας τρόπος πρόβλεψης των αριθμών που θα κληρωθούν στο ΛΟΤΤΟ είναι ο εξής: Δημιουργούμε ένα αρχείο με τις συχνότητες που κληρώθηκαν οι αριθμοί σε προηγούμενες κληρώσεις. Στη συνέχεια θεωρούμε ότι οι 6 πιο πιθανοί αριθμοί είναι αυτοί με τη μικρότερη συχνότητα εμφάνισης.

Να γραφεί αλγόριθμος που να διαβάζει το πλήθος των κληρώσεων, τους αριθμούς που κληρώθηκαν σε αυτές τις κληρώσεις και στη συνέχεια να εμφανίζει τους 6 πιο πιθανούς αριθμούς για την επόμενη κλήρωση.

16 Ο Kiss FM είχε κάνει για το Millennium τον εξής διαγωνισμό: έπαιρνε τηλέφωνο κάποιος ακροατής και του έλεγαν 6 αριθμούς, εκ των οποίων οι τέσσερις από αυτούς είχαν άθροισμα ακριβώς 2000. Ο ακροατής θα έπρεπε να βρει ποιοί είναι οι δύο αριθμοί που πρέπει να βγουν εκτός, έτσι ώστε το άθροισμα των υπολοίπων να είναι 2000.

ΤΡΙΚ

Να γραφεί αλγόριθμος ο οποίος θα διαβάζει έξι τέτοιους αριθμούς και θα βρίσκει ποιούς δύο πρέπει να εξαιρέσουμε, ώστε οι υπόλοιποι τέσσερις να έχουν άθροισμα 2000.

17 Σε μια αθλητική διοργάνωση συμμετέχουν 16 αθλητές του μήκους. Στην προκριματική φάση ο καθένας έχει δικαίωμα να κάνει 6 άλματα. Στην τελική φάση περνάνε όσοι κάνουν άλμα μεγαλύτερο των 8m. Να γράψετε αλγόριθμο ο οποίος:

- διαβάζει όλα τα άλματα, όλων των αθλητών
- υπολογίζει το μέγιστο άλμα του κάθε αθλητή
- υπολογίζει πόσοι αθλητές πέρασαν στην τελική φάση.

18 Στο NBA υπάρχει ο όρος Triple Double Figure ο οποίος σημαίνει ότι κάποιος παίκτης πέτυχε τρία διψήφια στατιστικά. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει έναν πίνακα της μορφής:

	Παίκτης 1	Παίκτης 2	Παίκτης 3	....	Παίκτης 20
Πόντοι				....	
Κλεψίματα				....	
Κοψίματα				....	
Assist				....	
Rebound				....	

Στη συνέχεια θα πρέπει να υπολογίζει πόσοι παίκτες πέτυχαν Triple Double Figure.

19 Ένας καθηγητής πληροφορικής κρατάει στατιστικά για τους 90 μαθητές του. Σε έναν πίνακα με το όνομα *Βαθμός* καταχωρεί τους βαθμούς που έγραψαν οι μαθητές του στο διαγώνισμα. Τα στατιστικά που τον ενδιαφέρουν είναι τα ακόλουθα:

- Ποσοστό μαθητών που έγραψε από 18 και πάνω.
- Ποσοστό μαθητών που έγραψε από 15 έως 18.
- Ποσοστό μαθητών που έγραψε από 10 έως 15.
- Ποσοστό μαθητών που έγραψε από 0 έως 10.

Να γραφεί αλγόριθμος ο οποίος διαβάζει τα στοιχεία του πίνακα και υπολογίζει τα παραπάνω στατιστικά.

20 Έστω ο ακόλουθος πίνακας με όνομα ΘΕΡΜΟΚΡΑΣΙΑ, ο οποίος περιέχει τις μέσες θερμοκρασίες που σημειώθηκαν σε τέσσερις πόλεις τους 12 μήνες του χρόνου:

	Αθήνα	Θεσσαλονίκη	Λάρισα	Βόλος
Ιανουάριος				
Φευρουάριος				
Μάρτιος				
....	....	....	....	....
Δεκέμβριος				

Τα ονόματα των πόλεων και των μηνών είναι καταχωρημένα στους πίνακες ΠΟΛΗ και ΜΗΝΑΣ αντίστοιχα. Να γραφεί αλγόριθμος ο οποίος:

- Διαβάζει τα στοιχεία του πίνακα ΘΕΡΜΟΚΡΑΣΙΑ.
- Υπολογίζει και εμφανίζει τη μέγιστη θερμοκρασία, το μήνα στον οποίο σημειώθηκε και σε ποια πόλη.
- Υπολογίζει και εμφανίζει τη μέση τιμή της θερμοκρασίας στη Λάρισα για όλο το χρόνο.

21 Να γραφεί αλγόριθμος ο οποίος βγάζει αποτελέσματα για το ΠΡΟ-ΠΟ. Δεδομένος είναι ο πίνακας ΝΙΚΗΤΡΙΑ ο οποίος περιέχει τη νικήτρια στήλη. Ο αλγόριθμος θα πρέπει:

- να διαβάζει τη στήλη που έπαιξε κάποιος παίκτης και να την καταχωρεί στον πίνακα ΠΡΟΒΛΕΨΗ,
- να υπολογίζει και να εμφανίζει το πλήθος των αγώνων που «έπιασε» ο παίκτης, και
- να εμφανίζει το μήνυμα «ΣΥΓΧΑΡΗΤΗΡΙΑ», αν ο παίκτης «πιάσει» απλό 13άρι, ενώ να εμφανίζει το μήνυμα «ΣΥΓΧΑΡΗΤΗΡΙΑ – ΣΟΥΠΕΡ 13» αν ο παίκτης «πιάσει» και το Σ13.

22 Να γραφεί αλγόριθμος, ο οποίος θα συγκρίνει δύο δεδομένους πίνακες Α και Β με Ν γραμμές και Μ στήλες ο καθένας. Αν οι πίνακες είναι ίσοι (έχουν τα στοιχεία τους ίσα ένα προς ένα), θα εμφανίζει το μήνυμα «ΙΣΟΙ».

23 Να γραφεί αλγόριθμος που να κατασκευάζει έναν πίνακα Α με την προπαίδεια. Η μορφή του Α θα πρέπει να είναι τέτοια, ώστε η πρώτη στήλη του να είναι η προπαίδεια του 1, η δεύτερη στήλη του να είναι η προπαίδεια του 2, κ.ο.κ.

24 Σε ένα σύγχρονο θερμοκήπιο 8 διαφορετικών χώρων υπάρχει ένα σύστημα μέτρησης και καταγραφής της θερμοκρασίας μέσω ενός υπολογιστή. Η μέση θερμοκρασία για κάθε ημέρα, κάθε δωματίου, καταχωρείται ανά γραμμή σ' έναν πίνακα που έχει 8 στήλες. Έτσι μετά την πάροδο ενός μήνα (30 ημέρες) δημιουργείται στον υπολογιστή ο πίνακας temp[30,8]. Να γίνει αλγόριθμος, ο οποίος θα δέχεται ως δεδομένα τον πίνακα temp και θα κάνει τα παρακάτω:

α) Να υπολογίζει και να εμφανίζει τη μέση θερμοκρασία ανά ημέρα.

β) Να υπολογίζει και να εμφανίζει τη μέση θερμοκρασία κάθε δωματίου για την περίοδο ενός μήνα

γ) Αν η μέση θερμοκρασία ενός δωματίου είναι μικρότερη από 20°C να εμφανίζει το μήνυμα «ΠΡΟΣΟΧΗ: ΧΑΜΗΛΗ ΘΕΡΜΟΚΡΑΣΙΑ» ενώ πάνω από τους 28°C να εμφανίζει το μήνυμα «ΠΡΟΣΟΧΗ: ΥΨΗΛΗ ΘΕΡΜΟΚΡΑΣΙΑ».

25 Σε ένα φροντιστήριο οι καθηγητές αποφάσισαν να δίνουν στους μαθητές, ως επιβράβευση, κάποια δώρα που η αξία τους υπολογίζεται ως ποσοστό επί των μηνιαίων διδάκτρων των μαθητών, ανάλογα με τον μέσο όρο των διαγωνισμάτων τους στο φροντιστήριο βάσει του ακόλουθου πίνακα:

Μέσος όρος διαγωνισμάτων	Ποσοστό δώρου
Μεγαλύτερος από 90	15%
Μεταξύ 85 και 90	10%
Μεταξύ 80 και 85	5%

Να αναπτύξετε αλγόριθμο, ο οποίος από τους πίνακες ΟΝΟΜΑΤΑ που περιέχει τα ονόματα των μαθητών, ΠΟΣΟ που περιέχει τα μηνιαία δίδακτρα του κάθε μαθητή και τον πίνακα ΜΟ που περιέχει τον μέσο όρο των διαγωνισμάτων του κάθε μαθητή, να εμφανίζει το όνομα του μαθητή και το ποσό που θα διαθέσουν οι καθηγητές για το δώρο του μαθητή.

26 Να γραφεί αλγόριθμος ο οποίος διαβάζει τις ηλικίες 100 ανθρώπων και τις καταχωρεί σε έναν μονοδιάστατο πίνακα Α. Κατόπιν υπολογίζει και εμφανίζει:

- Το μέσο όρο όλων των ηλικιών.
- Τη μέγιστη ηλικία.
- Το πλήθος των ανθρώπων που είναι άνω των 50 ετών.

27 Δίνεται ένας μονοδιάστατος πίνακας ο οποίος περιέχει τις βαθμολογίες 50 μαθητών σε κάποιο μάθημα. Κάποιος μαθητής θεωρείται ότι απέτυχε στο συγκεκριμένο μάθημα αν ο βαθμός του είναι μικρότερος του 9.5, ενώ κάποιος μαθητής θεωρείται ότι αρίστευσε αν ο βαθμός του είναι μεγαλύτερος ή ίσος του 19. Να γραφεί αλγόριθμος, ο οποίος θα διαβάζει τα στοιχεία ενός τέτοιου πίνακα και θα υπολογίζει το πλήθος των μαθητών που απέτυχαν και το πλήθος των μαθητών που αρίστευσαν.

28. Να γραφεί αλγόριθμος ο οποίος:

α) Να διαβάζει τα ονόματα και τις ετήσιες εισπράξεις 20 καταστημάτων.

β) Να εμφανίζει το πλήθος και τα ονόματα των καταστημάτων που έχουν εισπράξεις μικρότερες των 30.000 ευρώ.

γ) Να εμφανίζει το όνομα του καταστήματος με τις υψηλότερες εισπράξεις.

29. Ένα καταστηματάρχης χρειάζεται ένα πρόγραμμα για διαχείριση αποθήκης. Συγκεκριμένα, χρειάζεται ένα πρόγραμμα στο οποίο να μπορεί να αποθηκεύσει τα 200 προϊόντα του μαζί με τον κωδικό τους και τις διαθέσιμες ποσότητές τους. Έτσι, πληκτρολογώντας τον κωδικό, να μπορεί να βρει το όνομα του προϊόντος και τη διαθέσιμη ποσότητά του. Θεωρώντας δεδομένους τους πίνακες *ΚΩΔΙΚΟΣ*, *ΠΡΟΪΟΝ* και *ΠΟΣΟΤΗΤΑ*, οι οποίοι περιέχουν τους κωδικούς, τα ονόματα των προϊόντων και τις διαθέσιμες ποσότητές τους αντίστοιχα, να γράψετε τον σχετικό αλγόριθμο. Η μορφή των πινάκων είναι τέτοια που για παράδειγμα, το στοιχείο ΚΩΔΙΚΟΣ[5] είναι ο κωδικός του προϊόντος ΠΡΟΪΟΝ[5] του οποίου η διαθέσιμη ποσότητα είναι ΠΟΣΟΤΗΤΑ[5].

30. Να γραφούν αλγόριθμοι που να κατασκευάζουν τους παρακάτω πίνακες (έναν αλγόριθμο για κάθε πίνακα):

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 0 \\ 2 & 2 & 2 & 2 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \\ 49 \\ 64 \end{bmatrix}, \quad A = \begin{bmatrix} 4 \\ 8 \\ 12 \\ 16 \\ 20 \\ 24 \\ 28 \\ 32 \end{bmatrix}$$

31 Δίνεται ο πίνακας ακεραίων ΑΡΙΘΜΟΙ[100]. Να γραφεί αλγόριθμος που να διαχωρίζει τον πίνακα σε άρτιους και περιττούς (δηλαδή να δημιουργήσετε δύο νέους πίνακες).

32 Να γραφεί αλγόριθμος που να συγχωνεύει του πίνακες ΘΕΡΜ1[28] και ΘΕΡΜ2[32] σε έναν πίνακα.

33 Τέλειος λέγεται ο αριθμός που το άθροισμα των γνήσιων θετικών διαιρετών του είναι ίσο με τον αριθμό. Για παράδειγμα:

$$28 = 1+2+4+7+14$$

Με δεδομένο τον πίνακα ακεραίων ΑΡΙΘΜΟΙ[50], να κάνετε αλγόριθμο που θα εμφανίζει ποια στοιχεία του πίνακα είναι τέλειοι αριθμοί.