

ΚΕΦΑΛΑΙΟ 3:ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ

ΘΕΩΡΙΑ

3.2.Αλγόριθμοι + Δομές Δεδομένων=Προγράμματα

Τα δεδομένα ενός προβλήματος αποθηκεύονται στον υπολογιστή, είτε στην κύρια μνήμη του είτε στη δευτερεύουσα μνήμη του. Η αποθήκευση αυτή δεν γίνεται κατά ένα τυχαίο τρόπο αλλά συστηματικά, δηλαδή χρησιμοποιώντας μία δομή.

Δομή δεδομένων : Είναι ένα σύνολο αποθηκευμένων δεδομένων με συγκεκριμένο τρόπο που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

Η δομή αποτελείται από ένα σύνολο *κόμβων*. Κάθε κόμβος περιέχει μία απλή τιμή (αριθμό, αλφαριθμητικό κλπ) ή μία πιο σύνθετη τιμή (π.χ. μία εγγραφή)

Βασικές λειτουργίες (πράξεις) επί των δομών :

Προσπέλαση (Access)	Είναι η πρόσβαση σε ένα κόμβο με σκοπό την ανάγνωση ή τροποποίηση του περιεχομένου του.
Αναζήτηση (Searching)	Προσπελαύνονται οι κόμβοι μίας δομής με σκοπό να εντοπιστεί κάποιος που περιέχει μία συγκεκριμένη τιμή.
Εισαγωγή (Insertion)	Προσθήκη νέων κόμβων στη δομή.
Διαγραφή (Deletion)	Αντίστροφο της εισαγωγής - Αφαίρεση κόμβων από τη δομή.
Ταξινόμηση (Sorting)	Διατάσσουμε τους κόμβους της δομής σε αύξουσα ή φθίνουσα σειρά (π.χ. αλφαβητικά κατά επώνυμο, όνομα)
Αντιγραφή (Copying)	Όλοι ή μερικοί κόμβοι αντιγράφονται σε μία άλλη δομή
Συγχώνευση (Merging)	Δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
Διαχωρισμός (Separation)	Μία δομή διασπάται σε δύο ή περισσότερες. Είναι το αντίστροφο της συγχώνευσης.

Στην πράξη σπάνια χρησιμοποιούνται και οι οκτώ λειτουργίες για κάποια δομή. Συνηθέστε παρατηρείται το φαινόμενο μία δομή δεδομένων να είναι αποδοτικότερη από μία άλλη δομή με κριτήριο κάποια λειτουργία, για παράδειγμα την αναζήτηση, αλλά λιγότερο αποδοτική για κάποια άλλη λειτουργία, π.χ. την εισαγωγή. Τονίζεται ότι υπάρχει μεγάλη εξάρτηση ανάμεσα στη δομή δεδομένων και στον αλγόριθμο που επεξεργάζεται τη δομή αυτή. Η παρατήρηση αυτή οδήγησε τον Wirth (υλοποιητής της γλώσσας Pascal) στην εξής διαπίστωση:

Δομές δεδομένων + Αλγόριθμοι = Προγράμματα

Παράδειγμα: Ζητείται να υλοποιηθεί η κατάλληλη δομή δεδομένων και ο κατάλληλος αλγόριθμος αναζήτησης του τηλεφωνικού αριθμού ενός συνδρομητή του ΟΤΕ, δεδομένου του ονόματός του.

Λύση 1^η : Αποθηκεύουμε κατά τυχαίο τρόπο πρώτα το όνομα και μετά το τηλέφωνο του συνδρομητή ως εξής:

Όνομα 1 Τηλέφωνο1
Όνομα 2 Τηλέφωνο2
Όνομα 3 Τηλέφωνο3

.....

Ο παραπάνω τρόπος οργάνωσης των δεδομένων μας (δομή δεδομένων) είναι προφανές ότι δεν είναι αποδοτικός ειδικά για μεγάλο πλήθος συνδρομητών.

Λύση 2^η : Αποθηκεύουμε τα δεδομένα μας κατά αλφαβητική σειρά του ονόματος και επιπλέον σε μία δεύτερη δομή αποθηκεύουμε για κάθε γράμμα της αλφαβήτου τον αριθμό κόμβου μέσα στη δομή, που περιέχει τον 1^ο συνδρομητή που το όνομά του ξεκινά από το συγκεκριμένο γράμμα:

<u>1^η Δομή Δεδομένων</u>			<u>2^η Δομή Δεδομένων</u>	
1	Αναστασιάδης Δημήτριος	210-2728290	Α	1
2	Ανδρέου Γρηγόριος	210-3298789	Β	129
3	Αντύπας Γεώργιος	210-2526643	Γ	278
.....				
129	Βαγενάς Στυλιανός	210-4125675		
130	Βαρελάς Ιωάννης	210-9876009		
.....				
278	Γαρυφάλλου Ελένη	210-8721117		
279	Γιαννιάς Κώστας	210-7623006		

Με αυτό τον τρόπο είναι φανερό ότι είναι πιο εύκολη και ταχύτερη η αναζήτηση για κάποιον συνδρομητή.

Κατηγορίες δομών δεδομένων.

Στατικές	Δυναμικές
Το μέγεθός τους είναι καθορισμένο (σταθερό) και δεν μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. Το ακριβές μέγεθός τους καθορίζεται κατά την στιγμή της μετάφρασης του προγράμματος και όχι κατά την εκτέλεσή του.	Το μέγεθός τους δεν είναι καθορισμένο (δεν είναι σταθερό) και μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. Μπορούν να εισάγονται νέοι κόμβοι και να διαγράφονται υπάρχοντες. Στηρίζονται στην τεχνική της <i>δυναμικής παραχώρησης μνήμης</i> .
Οι κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.	Οι κόμβοι δεν αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.

3.3.Πίνακες

Χαρακτηριστικός εκπρόσωπος των στατικών δομών. Περιέχει ένα σταθερό σύνολο κόμβων (θέσεις) που αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Επίσης, όλα τα στοιχεία είναι του ίδιου τύπου (δηλαδή ακέραιοι, πραγματικοί κλπ).

α) Μονοδιάστατοι πίνακες (μίας γραμμής ή μίας στήλης).

Π.χ.

πίνακας ακεραίων Π [5]

Πίνακας αλφαριθμητικών Α [4]

12	-4	22	45	-7
1	2	3	4	5

“Γιάννης”	“Αλέκα”	“Μάριος”	“Άκης”
1	2	3	4

Οι αριθμοί κάτω από τις θέσεις του πίνακα δηλώνουν τον αριθμό θέσης του πίνακα. Ο αριθμός θέσης λέγεται και **δείκτης**.

Για να αναφερθούμε στο περιεχόμενο μία θέσης του πίνακα βάζουμε το **όνομα[θέση]**.

Για παράδειγμα, Π [1] = το περιεχόμενο της θέσης 1, Π [6] = το περιεχόμενο της θέσης 6

β) Διοδιάστατοι πίνακες (πολλών γραμμών και στηλών).

Εδώ έχουμε περισσότερες από μία γραμμές. Κάθε γραμμή έχει ένα σύνολο θέσεων (στήλες).

Στο παρακάτω παράδειγμα ο πίνακας ορίζεται ως εξής : ακέραιος T[3, 4]. Η πρώτη διάσταση αναφέρεται στις γραμμές και η δεύτερη στις στήλες.

Πίνακας T[3,4] ακεραίων:

	1	2	3	4
1	15	20	31	-6
2	61	8	-11	44
3	18	23	26	-4

Για να αναφερθούμε στο περιεχόμενο μία θέσης του διοδιάστατου πίνακα βάζουμε το **όνομα[γραμμή, στήλη]**. Δηλαδή, η θέση προσδιορίζεται από τον αριθμό γραμμής και στήλης. Για παράδειγμα, T [1,2] = το περιεχόμενο της θέσης στη γραμμή 1 και στήλη 2, T [3,2] = το περιεχόμενο της θέσης στη γραμμή 3 και στήλη 2.

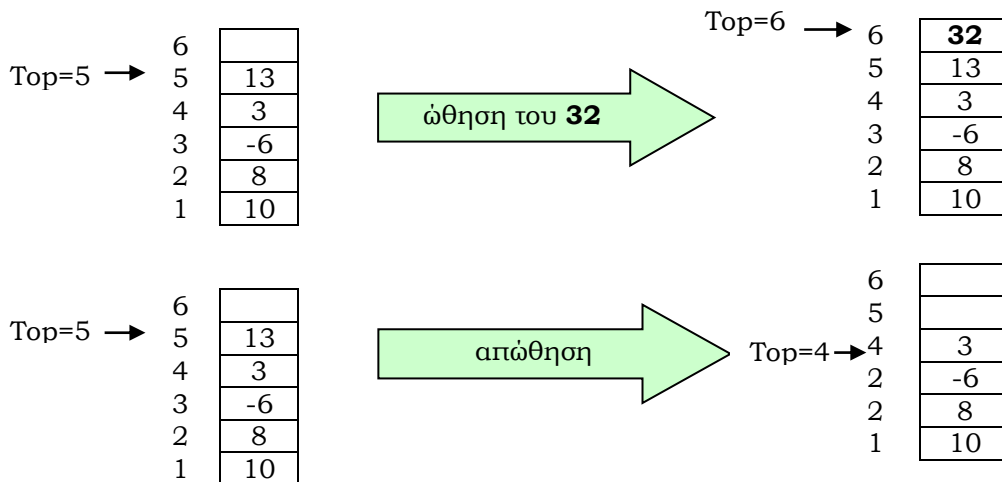
3.4.Στοιίβα (Stack).

Η στοιίβα υλοποιεί τη λογική **LIFO** (Last In First Out). Τα δεδομένα εισάγονται στην κορυφή της στοιίβας ενώ η αφαίρεση ενός στοιχείου γίνεται πάντα από την κορυφή της στοιίβας. Η **στατική** στοιίβα υλοποιείται με μονοδιάστατο πίνακα και χρησιμοποιεί ένα **δείκτη Top** που δείχνει στην κορυφή της στοιίβας. Όταν πρόκειται να εισαχθεί ένα στοιχείο στη στοιίβα, αυξάνεται ο δείκτης top κατά 1 και εισάγεται το στοιχείο στη θέση που δείχνει ο δείκτης top. Αντίστοιχα, όταν εξάγεται ένα στοιχείο, ο δείκτης top μειώνεται κατά 1 για να δείξει την νέα κορυφή της στοιίβας.

Οι λειτουργίες είναι δύο :

- Η **εισαγωγή-ώθηση (Push)** ενός στοιχείου στην κορυφή της στοιίβας.
- Η **αφαίρεση-απώθηση (Pop)** ενός στοιχείου από την κορυφή της στοιίβας.

Παράδειγμα :



Σημείωση : Κατά την εισαγωγή ενός στοιχείου (ώθηση), πρέπει να γίνεται έλεγχος μήπως η στοιίβα είναι γεμάτη (δηλαδή δεν υπάρχει άλλη ελεύθερη θέση στον πίνακα). Δηλαδή, ελέγχει μήπως συμβεί **υπερχείλιση (overflow)**. Αντίστοιχα, κατά την αφαίρεση ενός στοιχείου (απώθηση), πρέπει να γίνεται έλεγχος μήπως η στοιίβα είναι άδεια. Δηλαδή, ελέγχει μήπως συμβεί **υποχείλιση (underflow)**.

Οι εντολές που υλοποιούν τις λειτουργίες της ώθησης και απώθησης σε μία στοιίβα που υλοποιείται με έναν πίνακα A, N θέσεων είναι οι παρακάτω (ο δείκτης **top** σε μία στοιίβα N θέσεων μπορεί να πάρει τιμές από 0 (στοίβα άδεια) έως N(στοίβα γεμάτη)):

<u>ΩΘΗΣΗ</u>	<u>ΑΠΩΘΗΣΗ</u>
ΔΙΑΒΑΣΕ x ΑΝ top<N ΤΟΤΕ top← top+1 A[top]←x ΑΛΛΙΩΣ ΕΜΦΑΝΙΣΕ 'Υπερχείλιση' ΤΕΛΟΣ_ΑΝ	ΑΝ top>0 ΤΟΤΕ ΕΜΦΑΝΙΣΕ A[top] top← top-1 ΑΛΛΙΩΣ ΕΜΦΑΝΙΣΕ 'Υποχείλιση' ΤΕΛΟΣ_ΑΝ

3.5.Ουρά (Queue).

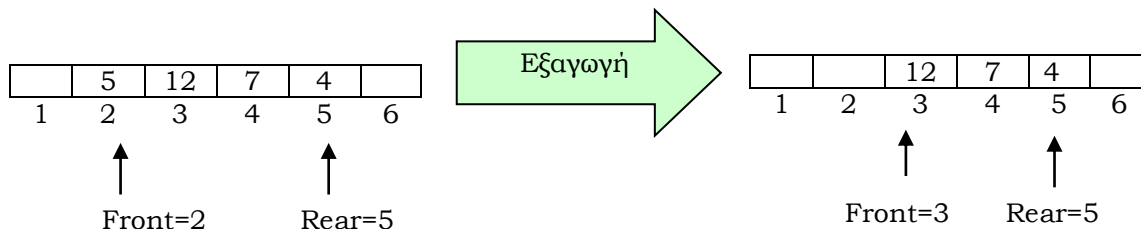
Η ουρά υλοποιεί τη λογική **FIFO** (First In First Out). Τα δεδομένα εισάγονται στο πίσω μέρος της ουράς ενώ η εξαγωγή ενός στοιχείου γίνεται πάντα από το μπροστινό μέρος της ουράς (όπως συμβαίνει σε μία ουρά σε τράπεζα!). Η **στατική ουρά** υλοποιείται με μονοδιάστατο πίνακα και χρησιμοποιεί δύο δείκτες : Τον **εμπρός(front)** που δείχνει το μπροστινό μέρος της ουράς και τον **πίσω(rear)** που δείχνει το πίσω μέρος της ουράς. Όταν εισάγεται ένα στοιχείο στην ουρά αυξάνεται ο δείκτης **πίσω** κατά 1 και εισάγεται το στοιχείο στη θέση που δείχνει ο δείκτης **πίσω**, ενώ όταν εξάγεται ένα στοιχείο αυξάνεται ο δείκτης **εμπρός** κατά 1.

Οι λειτουργίες λοιπόν είναι δύο :

- Η **εισαγωγή** ενός στοιχείου στο πίσω μέρος της ουράς.
- Η **εξαγωγή** ενός στοιχείου από το μπροστινό μέρος της ουράς.

Παράδειγμα :





Σημείωση : Κατά την εισαγωγή ενός στοιχείου, πρέπει να γίνεται έλεγχος μήπως η ουρά είναι γεμάτη (δηλαδή δεν υπάρχει άλλη ελεύθερη θέση στον πίνακα). Αντίστοιχα, κατά την εξαγωγή ενός στοιχείου, πρέπει να γίνεται έλεγχος μήπως η ουρά είναι άδεια. Επίσης, όταν $front=rear$ τότε η ουρά έχει ένα στοιχείο, ενώ όταν $front>rear$ ($front=rear+1$) τότε η ουρά δεν έχει κανένα στοιχείο.

Οι εντολές που υλοποιούν τις λειτουργίες της εισαγωγής και εξαγωγής σε μία ουρά που υλοποιείται με έναν πίνακα A, N θέσεων είναι οι παρακάτω (ο δείκτης **πίσω** σε μία ουρά N θέσεων μπορεί να πάρει τιμές από 0 (πριν μπει το πρώτο στοιχείο) έως N(ουρά γεμάτη). Ο δείκτης **εμπρός** έχει ελάχιστη τιμή το 1(πριν μπει το πρώτο στοιχείο) και μέγιστη τιμή το N+1 (όταν η ουρά αδειάσει)):

<u>ΕΙΣΑΓΩΓΗ</u>	<u>ΕΞΑΓΩΓΗ</u>
ΔΙΑΒΑΣΕ x ΑΝ πίσω<N ΤΟΤΕ πίσω← πίσω+1 A[πίσω]←x ΑΛΛΙΩΣ ΕΜΦΑΝΙΣΕ 'Ουρά γεμάτη' ΤΕΛΟΣ_ΑΝ	ΑΝ εμπρός≤πίσω ΤΟΤΕ ΕΜΦΑΝΙΣΕ A[εμπρός] εμπρός← εμπρός+1 ΑΛΛΙΩΣ ΕΜΦΑΝΙΣΕ 'Ουρά άδεια' ΤΕΛΟΣ_ΑΝ

3.6.Αναζήτηση.

Έστω ότι τίθεται το πρόβλημα της αναζήτησης ενός στοιχείου (μεταβλητή key) σε έναν μονοδιάστατο πίνακα. Υπάρχουν πλήθος αλγορίθμων αναζήτησης. Εδώ θα δούμε την **σειριακή μέθοδο** που αναζητεί το συγκεκριμένο στοιχείο σε έναν μη ταξινομημένο πίνακα διαφορετικών στοιχείων. Η λογική είναι η εξής : Σαρώνουμε τον πίνακα και εξετάζουμε σε κάθε θέση αν το στοιχείο της θέσης αυτής είναι ίσο με αυτό που ψάχνουμε. Αν ναι, τότε σταματάμε.

Θα χρειαστούμε μία μεταβλητή **βρέθηκε** λογικού τύπου, που θα γίνει Αληθής αν το στοιχείο βρεθεί. Επίσης, δεν γνωρίζουμε σε ποια θέση βρίσκεται. Θα χρησιμοποιήσουμε και μία μεταβλητή **θέση** που θα κρατήσει τη θέση όπου βρέθηκε το στοιχείο που ψάχνουμε.

Αλγόριθμος Σειριακή_αναζήτηση

Δεδομένα //N, Π, key //

βρέθηκε ← Ψευδής

θέση ← 0

i ← 1

Όσο (i ≤ N) **ΚΑΙ** (βρέθηκε = ψευδής) **επανάλαβε** *!όσο δεν είμαστε στο τέλος του πίνακα και δεν βρέθηκε*

Αν Π[i] = key **τότε**

 βρέθηκε ← Αληθής

 θέση ← i

αλλιώς

 i ← i + 1

τέλος_αν

τέλος_επανάληψης

Αν βρέθηκε = Αληθής **τότε**

Τύπωσε "Το στοιχείο βρέθηκε στη θέση ", θέση

αλλιώς

Τύπωσε "Το στοιχείο δεν βρέθηκε "

τέλος_αν

Τέλος Σειριακή_αναζήτηση

!αν το i στοιχείο ισούται με αυτό που ψάχνουμε τότε

!βρέθηκε

!βάλε τον αριθμό θέσης στην μεταβλητή θέση

!αλλιώς

!προχώρα στην επόμενη θέση

Πότε χρησιμοποιούμε τη μέθοδο αυτή:

- Όταν ο πίνακας είναι μη ταξινομημένος.
- Ο πίνακας είναι μικρού μεγέθους (π.χ. N ≤ 20).
- Η αναζήτηση να πραγματοποιείται σπάνια, διότι η μέθοδος αυτή είναι σχετικά αργή.

Σημείωση : Όταν ο πίνακας είναι ταξινομημένος είναι προτιμότερο να χρησιμοποιείται μία πιο αποδοτική (γρήγορη) μέθοδος που ονομάζεται δυναμική αναζήτηση.

Παραλλαγή της μεθόδου της σειριακής αναζήτησης σε πίνακα που κάθε στοιχείο δεν υπάρχει μόνο μία φορά:

Αλγόριθμος Σειριακή_αναζήτηση1

Δεδομένα //N, Π, key //

θέση $\leftarrow 0$

Για i από 1 μέχρι N

Αν Π[i] = key **τότε**

Τύπωσε " Το στοιχείο βρέθηκε στη θέση ", i

 θέση $\leftarrow i$

τέλος_αν

τέλος_επανάληψης

Αν θέση=0 **τότε**

Τύπωσε "Το στοιχείο δεν βρέθηκε "

τέλος_αν

Τέλος Σειριακή_αναζήτηση1

Παραλλαγή της μεθόδου της σειριακής αναζήτησης σε ταξινομημένο πίνακα (αύξουσα ταξινόμηση):

Αλγόριθμος Σειριακή_αναζήτηση2

Δεδομένα //N, Π, key //

Stop $\leftarrow$ Ψευδής

θέση $\leftarrow 0$

i $\leftarrow 1$

Όσο (i <= N) **ΚΑΙ** (stop = ψευδής) **επανάλαβε**

Αν Π[i] = key **τότε**

 stop $\leftarrow$ Αληθής

 θέση $\leftarrow i$

αλλιώς_αν Π[i]>key **τότε**

 stop $\leftarrow$ Αληθής

αλλιώς

 i $\leftarrow i + 1$

τέλος_αν

τέλος_επανάληψης

Αν θέση<>0 **τότε**

Τύπωσε "Το στοιχείο βρέθηκε στη θέση ", θέση

αλλιώς

Τύπωσε "Το στοιχείο δεν βρέθηκε "

τέλος_αν

Τέλος Σειριακή_αναζήτηση2

3.7. Ταξινόμηση.

Η τακτοποίηση των κόμβων μίας δομής με μια ιδιαίτερη σειρά είναι μία πολύ σημαντική λειτουργία που ονομάζεται ταξινόμηση. Σκοπός της ταξινόμησης είναι να διευκολυνθεί στη συνέχεια η αναζήτηση των στοιχείων του ταξινομημένου πίνακα. Στην πράξη η ταξινόμηση χρησιμοποιείται σε περιπτώσεις αναζήτησης δεδομένων όπως σε βιβλιοθηκονομικά συστήματα, λεξικά, τηλεφωνικούς καταλόγους, κ.τ.λ.

Ορισμός: Δοθέντων στοιχείων **a1,a2,...,an** η ταξινόμηση συνίσταται στη μετάθεση της θέσης των στοιχείων ώστε να τοποθετηθούν σε μία σειρά **ak1,ak2,...,akn** ώστε δοθείσης μίας συνάρτησης διάταξης f, να ισχύει:

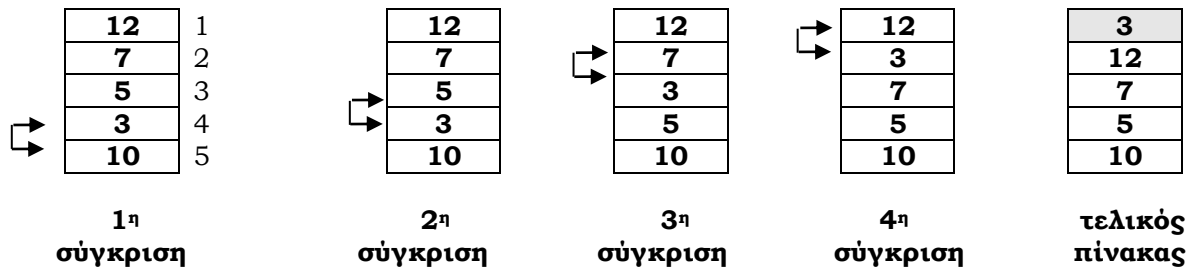
$$f(ak1) \leq f(ak2) \leq \dots \leq f(akn).$$

Ταξινόμηση ευθείας ανταλλαγής (ταξινόμηση φυσσαλίδας). Βασίζεται στην αρχή της σύγκρισης γειτονικών στοιχείων του πίνακα και ανταλλαγής τους μέχρι να διαταχθούν όλα σε μία σειρά (αύξουσα ή φθίνουσα). Η λογική είναι η εξής : Κάνουμε διαδοχικές σαρώσεις στον πίνακα. Σε κάθε σάρωση, το μικρότερο στοιχείο (για αύξουσα ταξινόμηση) μετακινείται προς την κορυφή του πίνακα. Αφού γίνουν όλες οι σαρώσεις θα έχει επιτευχθεί η ταξινόμηση.

Παράδειγμα : Να ταξινομηθούν οι ακέραιοι 12, 7, 5, 3, 10 ενός πίνακα με τη μέθοδο της φυσσαλίδας.

1^η σάρωση: Στην πρώτη σάρωση του πίνακα γίνονται 4 συγκρίσεις. Ξεκινάμε από τα δύο τελευταία στοιχεία και πηγαίνουμε προς τα πάνω. Δηλαδή πρώτα συγκρίνουμε το 5^ο με το 4^ο στοιχείο, μετά το 4^ο με το 3^ο ,

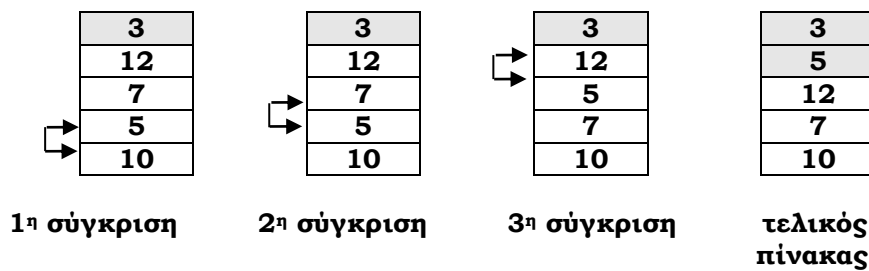
κ.ο.κ. Αν το από κάτω στοιχείο είναι μικρότερο, αντιμεταθέτουμε τα στοιχεία. Έτσι, στο τέλος της πρώτης σάρωσης το μικρότερο στοιχείο του πίνακα βρίσκεται στην πρώτη (πάνω) θέση.



Οι εντολές που υλοποιούν την πρώτη σάρωση θα είναι:

ΓΙΑ j ΑΠΟ 5 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ -1
ΑΝ $\Pi[j-1] > \Pi[j]$ ΤΟΤΕ
ΑΝΤΙΜΕΤΑΘΕΣΣΕ $\Pi[j-1]$, $\Pi[j]$
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

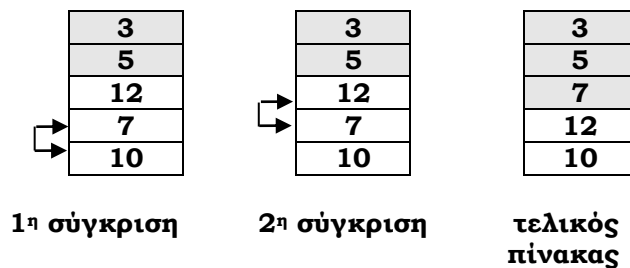
2^η σάρωση: Στη δεύτερη σάρωση θα γίνουν 3 συγκρίσεις ξεκινώντας πάλι από τα δύο τελευταία στοιχεία. Το δεύτερο μικρότερο στοιχείο του πίνακα θα έρθει στη δεύτερη θέση:



Οι εντολές που υλοποιούν την δεύτερη σάρωση θα είναι:

ΓΙΑ j ΑΠΟ 5 ΜΕΧΡΙ 3 ΜΕ_ΒΗΜΑ -1
ΑΝ $\Pi[j-1] > \Pi[j]$ ΤΟΤΕ
ΑΝΤΙΜΕΤΑΘΕΣΣΕ $\Pi[j-1]$, $\Pi[j]$
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

3^η σάρωση: Στη τρίτη σάρωση θα γίνουν 2 συγκρίσεις ξεκινώντας πάλι από τα δύο τελευταία στοιχεία. Το τρίτο μικρότερο στοιχείο του πίνακα θα έρθει στη τρίτη θέση:



Οι εντολές που υλοποιούν την τρίτη σάρωση θα είναι:

ΓΙΑ j ΑΠΟ 5 ΜΕΧΡΙ 4 ΜΕ_ΒΗΜΑ -1
ΑΝ $\Pi[j-1] > \Pi[j]$ ΤΟΤΕ
ΑΝΤΙΜΕΤΑΘΕΣΣΕ $\Pi[j-1]$, $\Pi[j]$
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

4^η σάρωση: Στη τέταρτη σάρωση θα συγκριθούν τα δύο τελευταία στοιχεία του πίνακα:



Οι εντολές που υλοποιούν την τέταρτη σάρωση θα είναι:

```

ΓΙΑ j ΑΠΟ 5 ΜΕΧΡΙ 5 ΜΕ_ΒΗΜΑ -1
  ΑΝ Π[j -1] >Π[j] ΤΟΤΕ
    ΑΝΤΙΜΕΤΑΘΕΣΕ Π[j -1] , Π[j]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Παρατηρούμε ότι οι εντολές σε κάθε σάρωση είναι ίδιες, εκτός της τελικής τιμής του μετρητή j ο οποίος μεταβάλλεται από 2 (στην 1^η σάρωση) μέχρι 5 (στην τελευταία σάρωση). Μπορούμε λοιπόν να γράψουμε τις παρακάτω εντολές για την ταξινόμηση ενός πίνακα 5 θέσεων με τη μέθοδο της φυσσαλίδας:

```

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 5
  ΓΙΑ j ΑΠΟ 5 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
    ΑΝ Π[j -1] >Π[j] ΤΟΤΕ
      ΑΝΤΙΜΕΤΑΘΕΣΕ Π[j -1] , Π[j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Και στην γενική περίπτωση για έναν πίνακα Π, N θέσεων να γράψουμε:

```

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N
  ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
    ΑΝ Π[j -1] >Π[j] ΤΟΤΕ
      ΑΝΤΙΜΕΤΑΘΕΣΕ Π[j -1] , Π[j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

ή

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N-1
  ΓΙΑ j ΑΠΟ N-1 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
    ΑΝ Π[j] >Π[j+1] ΤΟΤΕ
      ΑΝΤΙΜΕΤΑΘΕΣΕ Π[j],Π[j+1]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Παρατηρήσεις:

- Η εντολή ΑΝΤΙΜΕΤΑΘΕΣΕ μπορεί να χρησιμοποιηθεί μόνο σε αλγόριθμο. Αν χρειαστεί να χρησιμοποιηθεί ο αλγόριθμος της φυσσαλίδας σε πρόγραμμα, τότε αντί της εντολής αυτής θα πρέπει να γράψουμε:

```

temp←Π[j-1]      temp←Π[j]
Π[j-1]←Π[j]      ή   Π[j]←Π[j-1]
Π[j]←temp        Π[j-1]←temp

```

- Ο παραπάνω αλγόριθμος ισχύει για αύξουσα ταξινόμηση (από το μικρότερο στο μεγαλύτερο στοιχείο). Για φθίνουσα ταξινόμηση (από μεγαλύτερο σε μικρότερο στοιχείο) θα αλλάζαμε τη φορά της ανισότητας δηλ. **ΑΝ Π[j -1] < Π[j] ΤΟΤΕ**
- Κατά την ταξινόμηση γίνονται N-1 σαρώσεις στον πίνακα. Στην πρώτη σάρωση γίνονται N-1 συγκρίσεις και σε κάθε επόμενη σάρωση γίνεται μία σύγκριση λιγότερη. Στη τελευταία σάρωση γίνεται πάντα μία σύγκριση. Συνολικά γίνονται **N*(N-1)/2** συγκρίσεις.

Δομές δεδομένων δευτερεύουσας μνήμης.

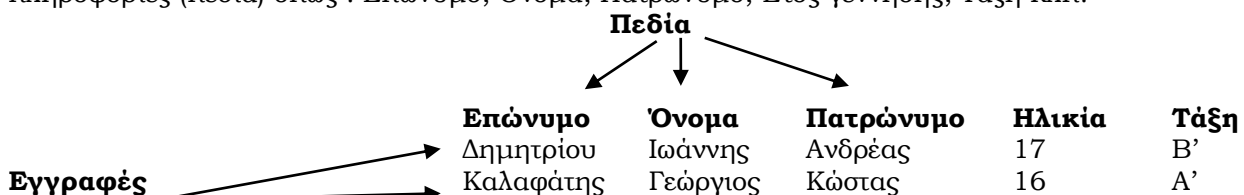
Επειδή η RAM δεν επαρκεί για την αποθήκευση των δεδομένων καθώς και το ότι δεν αποθηκεύει μόνιμα, χρησιμοποιούμε τους μαγνητικούς δίσκους (ταινίες, σκληροί δίσκοι κ.α.) και οπτικούς δίσκους (CD-ROM, DVD-ROM κ.α.). Αυτά αποτελούν την **περιφερειακή ή δευτερεύουσα μνήμη**.

Για την αποθήκευση δεδομένων σε αυτή κάνουμε χρήση ειδικών δομών που λέγονται **αρχεία (files)**.

Κάθε αρχείο περιέχει μία συλλογή εγγραφών (π.χ. ένα αρχείο με εγγραφές μαθητών).

Μία **εγγραφή (Record)** αποτελεί τη συλλογή στοιχειωδών πληροφοριών σχετικά με ένα αντικείμενο (π.χ. μαθητής, ένα βιβλίο κλπ).

Μία στοιχειώδης πληροφορία συνιστά ένα **πεδίο (field)**. Π.χ. μία εγγραφή μαθητή περιέχει στοιχειώδεις πληροφορίες (πεδία) όπως : Επώνυμο, Όνομα, Πατρώνυμο, Έτος γέννησης, Τάξη κλπ.



Το πεδίο που ταυτοποιεί μία εγγραφή (την κάνει μοναδική) ονομάζεται *πρωτεύον κλειδί*, π.χ. ο αριθμός μητρώου ενός μαθητή, ενώ το Ονοματεπώνυμο *δευτερεύον κλειδί* (αν υπήρχε το πρωτεύον κλειδί).

3.9 Άλλες δομές δεδομένων

Κοινό γνώρισμα των δομών δεδομένων που εξετάστηκαν προηγουμένως είναι ότι οι διαδοχικοί κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις της κύριας μνήμης. Στην παράγραφο αυτή γίνεται παρουσίαση τριών πολύ σπουδαίων δομών δεδομένων, στις οποίες οι κόμβοι δεν είναι απαραίτητο να κατέχουν συνεχόμενες θέσεις μνήμης. Είναι οι λίστες, τα δένδρα και οι γράφοι.

3.9.1 Λίστες

Στις λίστες το κύριο χαρακτηριστικό είναι ότι οι κόμβοι τους συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης και η σύνδεσή τους γίνεται με δείκτες². Ο δείκτης (pointer¹) είναι ένας ιδιαίτερος τύπος που προσφέρεται από τις περισσότερες σύγχρονες γλώσσες προγραμματισμού. Ο δείκτης δεν λαμβάνει αριθμητικές τιμές όπως ακέραιες, πραγματικές, κ.α., αλλά οι τιμές του είναι διευθύνσεις στην κύρια μνήμη και χρησιμοποιείται ακριβώς για τη σύνδεση των διαφόρων στοιχείων μιας δομής, που είναι αποθηκευμένα σε μη συνεχόμενες θέσεις μνήμης. Συνήθως ο δείκτης είναι ένα πεδίο κάθε κόμβου της δομής, όπως φαίνεται στο κάτωθι σχήμα. Το πεδίο Δεδομένα μπορεί να περιέχει μία ή περισσότερες αριθμητικές ή αλφαριθμητικές πληροφορίες.



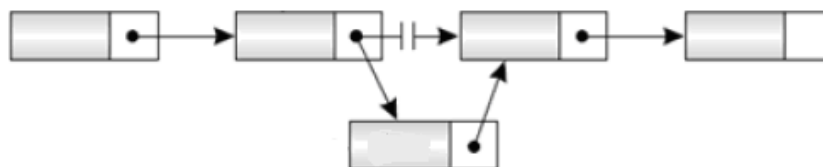
Δομή κόμβου.

Στο επόμενο σχήμα παρουσιάζεται μία λίστα με τέσσερις κόμβους, όπου οι δείκτες έχουν τη μορφή βέλους, προκειμένου να φαίνεται ο κόμβος που παραπέμπουν.



λίστα με τέσσερις κόμβους

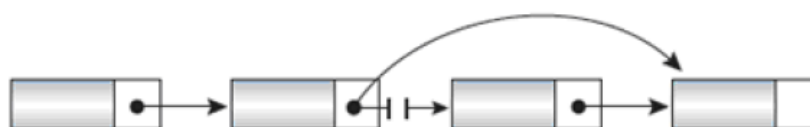
Με τη χρήση δεικτών διευκολύνονται οι λειτουργίες της εισαγωγής και της διαγραφής δεδομένων από τις λίστες. Στο επόμενο σχήμα φαίνεται η εισαγωγή ενός νέου κόμβου μεταξύ του δεύτερου και τρίτου κόμβου της προηγούμενης λίστας.



Εισαγωγή σε λίστα

Όπως φαίνεται και στο σχήμα, οι απαιτούμενες ενέργειες για την εισαγωγή (παρεμβολή) του νέου κόμβου είναι ο δείκτης του δεύτερου κόμβου να δείχνει το νέο κόμβο και ο δείκτης του νέου κόμβου να δείχνει τον τρίτο κόμβο (δηλαδή να πάρει την τιμή που είχε πριν την εισαγωγή ο δείκτης του δεύτερου κόμβου). Έτσι, οι κόμβοι της λίστας διατηρούν τη λογική τους σειρά, αλλά οι φυσικές θέσεις στη μνήμη μπορεί να είναι τελείως διαφορετικές.

Αντίστοιχα για τη διαγραφή ενός κόμβου αρκεί να αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον τον επόμενο αυτού που διαγράφεται όπως φαίνεται στο επόμενο σχήμα. Ο κόμβος που διαγράφηκε (ο τρίτος) αποτελεί “άχρηστο δεδομένο” και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.

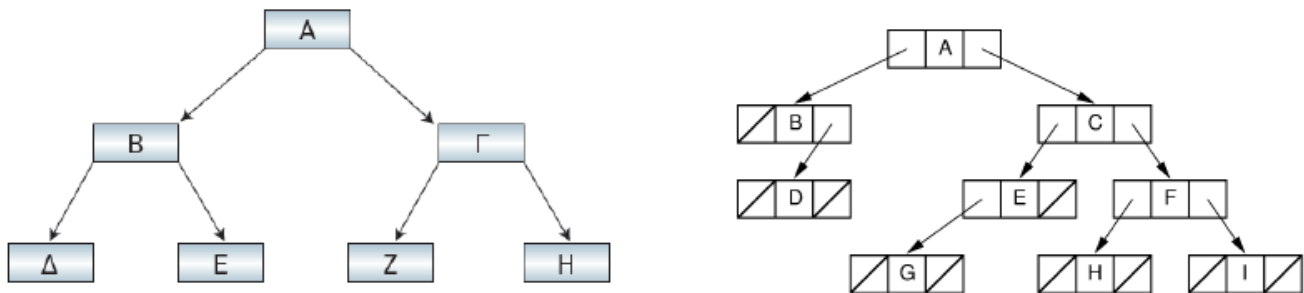


Διαγραφή κόμβου λίστας

1. Οι όροι *index* και *pointer* αποδίδονται στα ελληνικά ως δείκτης. Και οι δύο παραπέμπουν σε θέσεις, πίνακα ο πρώτος και μνήμης ο δεύτερος.
2. Οι δομές δεδομένων που χρησιμοποιούν δείκτες αποκαλούνται δυναμικές (*dynamic*), γιατί η υλοποίησή τους γίνεται έτσι, ώστε να μην απαιτείται εκ των προτέρων καθορισμός του μέγιστου αριθμού κόμβων. Είναι φανερό ότι οι δομές αυτές είναι πιο ευέλικτες από τη στατική δομή του πίνακα, επειδή επεκτείνονται και συρρικνώνονται κατά τη διάρκεια εκτέλεσης του προγράμματος.

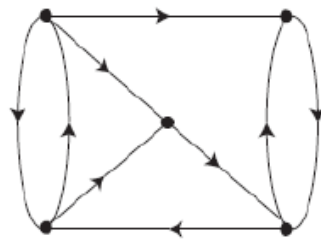
3.9.2 Δένδρα

Τα δένδρα (*trees*) είναι δομές που στις σύγχρονες γλώσσες προγραμματισμού υλοποιούνται με τη βοήθεια των δεικτών, όπως εξηγήθηκε πριν. Βέβαια, μπορούν να υλοποιηθούν και με στατικές δομές (με πίνακες). Το κύριο χαρακτηριστικό των δένδρων είναι, ότι από έναν κόμβο δεν υπάρχει μόνο ένας επόμενος κόμβος, αλλά περισσότεροι. Υπάρχει ένας μόνο κόμβος, που λέγεται ρίζα, από τον οποίο ξεκινούν όλοι οι άλλοι κόμβοι. Στο επόμενο σχήμα παρατηρούμε ότι από τη ρίζα ξεκινούν δύο κόμβοι. Οι κόμβοι αυτοί λέγονται παιδιά της ρίζας. Με την ίδια λογική, από κάθε παιδί της ρίζας ξεκινούν άλλα παιδιά, κ.ο.κ.



3.9.3 Γράφοι

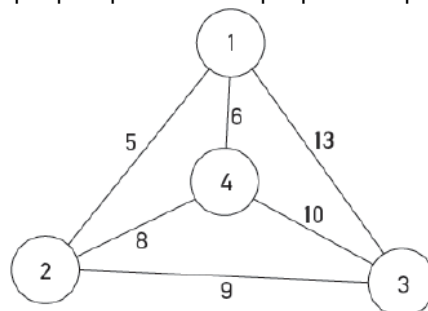
Ένας γράφος (*graph*) αποτελείται από ένα σύνολο κόμβων (ή σημείων ή κορυφών) και ένα σύνολο γραμμών (ή ακμών ή τόξων) που ενώνουν μερικούς ή όλους τους κόμβους. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι όλες οι προηγούμενες δομές που παρουσιάστηκαν μπορούν να θεωρηθούν περιπτώσεις γράφων.



Πολλά προβλήματα και καταστάσεις της καθημερινής μας ζωής μπορούν να περιγραφούν με τη βοήθεια γράφων. Για παράδειγμα, τα σημεία ενός γράφου μπορούν να παριστούν πόλεις και οι γραμμές τις οδικές συνδέσεις μεταξύ τους. Λόγω της μεγάλης πληθώρας και ποικιλίας των προβλημάτων που σχετίζονται με γράφους, έχει αναπτυχθεί ομώνυμη θεωρία, η Θεωρία Γράφων, η οποία συχνά αποτελεί αυτοδύναμο μάθημα σε πανεπιστημιακά τμήματα.

Π.χ. Πρόβλημα κινέζου ταχυδρόμου

Επιθυμούμε ο ταχυδρομικός διανομέας, να ξεκινήσει από ένα χωριό, να επισκεφθεί έναν αριθμό από γειτονικά χωριά, για να μοιράσει ένα σύνολο επιστολών και να επιστρέψει στο χωριό, από όπου ξεκίνησε περνώντας μόνο μία φορά από κάθε χωριό. Το πρόβλημα έγκειται στην εύρεση της καλύτερης διαδρομής, έτσι ώστε ο διανομέας να διανύσει το μικρότερο δυνατό αριθμό χιλιομέτρων.



ΑΣΚΗΣΕΙΣ

ΔΥΑΔΙΚΗ ΑΝΑΖΗΤΗΣΗ

Ο αλγόριθμος της δυαδικής αναζήτησης (binary search) εφαρμόζεται μόνο σε πίνακες που έχουν ταξινομημένα στοιχεία. Αν τα στοιχεία δεν είναι ταξινομημένα τότε δεν μπορεί να εφαρμοστεί.

Ο αλγόριθμος λειτουργεί ως εξής:

Βρίσκουμε το μεσαίο στοιχείο του ταξινομημένου πίνακα. Εάν το προς αναζήτηση στοιχείο είναι ίσο με το μεσαίο στοιχείο τότε σταματάμε την αναζήτηση αφού το στοιχείο βρέθηκε.

Εάν δεν βρέθηκε, τότε ελέγχουμε αν το στοιχείο που αναζητούμε είναι μικρότερο ή μεγαλύτερο από το μεσαίο στοιχείο του πίνακα. Αν είναι μικρότερο, περιορίζουμε την αναζήτηση στο πρώτο μισό του πίνακα (με την προϋπόθεση ότι τα στοιχεία είναι διατεταγμένα κατά αύξουσα σειρά), ενώ αν είναι μεγαλύτερο περιορίζουμε την αναζήτηση στο δεύτερο μισό του πίνακα. Η διαδικασία αυτή λοιπόν επαναλαμβάνεται για το κατάλληλο πρώτο ή δεύτερο μισό πίνακα, μετά για το 1/4 του πίνακα κ.ο.κ. μέχρι, είτε να βρεθεί το στοιχείο, είτε να μην είναι δυνατό να χωρισθεί ο πίνακας περαιτέρω σε δύο νέα μέρη.

Αλγόριθμος δυαδικής αναζήτησης

Αλγόριθμος Δυαδική αναζήτηση

δεδομένα // N, A, S //

!Α μονοδιάστατος πίνακας N θέσεων, S το στοιχείο που αναζητούμε

Left ← 1

! αριστερό όριο

Right ← N

! δεξιό όριο

K ← 0

! θέση του στοιχείου

F ← ΨΕΥΔΗΣ

όσο (Left ≤ Right) **και** (F = ΨΕΥΔΗΣ) **επανάλαβε**

M ← (Left + Right) div 2

αν A[M] = S **τότε**

K ← M

F ← ΑΛΗΘΗΣ

αλλιώς

αν A[M] < S **τότε**

Left ← M + 1;

αλλιώς

Right ← M - 1;

Τέλος_αν

Τέλος_αν

Τέλος_επανάληψης

Αν F = ΑΛΗΘΗΣ **τότε**

Εμφάνισε "Το στοιχείο,", S , "υπάρχει στη θέση:", M

Αλλιώς

Εμφάνισε "Το στοιχείο,", S , " δεν υπάρχει στον πίνακα"

Τέλος_αν

Τέλος Δυαδική αναζήτηση

Παράδειγμα

Δίνεται ο πίνακας A:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

α) Αναζήτηση του στοιχείου 38 (υπάρχει στον πίνακα)

Βήμα 1

Left						M	Right							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	2	5	8	9	15	22	25	27	35	37	38	40	43	

Left=1 Right=14 $M=(1+14) \div 2=7$
 $A[M]=A[7]=22 < 38$, άρα για το επόμενο βήμα Left=M+1=7+1=8

Βήμα 2

							Left			M		Right	
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

Left=8 Right=14 $M=(8+14) \div 2=11$
 $A[M]=A[11]=37 < 38$, άρα άρα για το επόμενο βήμα Left=M+1=11+1=12

Βήμα 3

											Left	M	Right
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

Left=12 Right=14 $M=(12+14) \div 2=13$
 $A[M]=A[13]=40 > 38$, άρα άρα για το επόμενο βήμα Right=M-1=13-1=12

Βήμα 4

											Left=Right=M		
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

Left=12 Right=12 $M=(12+12) \div 2=12$
 $A[M]=A[12]=38$, άρα τελειώνει η αναζήτηση και το στοιχείο βρέθηκε στη θέση 12.

β) Αναζήτηση του στοιχείου 10 (δεν υπάρχει στον πίνακα)

Βήμα 1

Left						M	Right							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	2	5	8	9	15	22	25	27	35	37	38	40	43	

Left=1 Right=14 $M=(1+14) \div 2=7$
 $A[M]=A[7]=22 > 10$, άρα για το επόμενο βήμα Right=M-1=7-1=6

Βήμα 2

Left			M			Right							
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

Left=1 Right=6 $M=(1+6) \div 2=3$
 $A[M]=A[3]=5 < 10$, άρα άρα για το επόμενο βήμα Left=M+1=3+1=4

Βήμα 3

			Left	M	Right								
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

Left=4 Right=6 $M=(4+6) \div 2=5$

$A[M]=A[5]=9 < 10$, άρα άρα για το επόμενο βήμα $Left=M+1=5+1=6$

Βήμα 4

				Left=Right=M									
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	5	8	9	15	22	25	27	35	37	38	40	43

Left=6 Right=6 $M=(6+6) \div 2=6$

$A[M]=A[6]=15 > 10$, άρα για το επόμενο βήμα $Right=6-1=5$.

Βήμα 5

Left=6 Right=5, άρα τερματίζεται η επανάληψη επειδή $Left > Right$. Το στοιχείο δεν βρέθηκε.

Αριθμός συγκρίσεων στη δυαδική αναζήτηση :

Στοιχεία N	Συγκρίσεις
10	4
100	7
1.000	10
10.000	14
100.000	17
1.000.000	20
10.000.000	24
100.000.000	27
1.000.000.000	30

Αποδεικνύεται ότι ο μέγιστος αριθμός συγκρίσεων στη δυαδική αναζήτηση είναι ένας θετικός ακέραιος έτσι ώστε να ισχύει: $2_{\text{αριθμός συγκρίσεων}} \geq N$ (π.χ. αν $N=10$, αριθμός συγκρίσεων=4). Ισχύει δηλαδή ο τύπος :

$$\text{αριθμός συγκρίσεων} = (A_M(\log_2 N) + 1)$$

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (SELECTION SORT)

Η ταξινόμηση με επιλογή (selection sort), αποτελεί βασικό τρόπο ταξινόμησης, που υλοποιείται σε ένα μονοδιάστατο πίνακα σε τρία βήματα:

1. Επιλογή του ελάχιστου στοιχείου
2. Ανταλλαγή του ελάχιστου με το πρώτο στοιχείο
3. Επανάληψη των βημάτων 1 και 2 για τα υπόλοιπα στοιχεία του πίνακα

Ο Αλγόριθμος ταξινόμησης με επιλογή είναι ο παρακάτω.

Αλγόριθμος Selection_Sort

Δεδομένα // table, n //

Για i από 1 μέχρι n-1

k ← i

x ← table[i]

Για j από i+1 μέχρι n

Αν x < table[j] **Τότε**

k ← j

x ← table[j]

Τέλος_αν

Τέλος_Επανάληψης

table[k] ← table[i]

table[i] ← x

Τέλος_ επανάληψης

Τέλος Selection_Sort

Παράδειγμα

Αν υποθέσουμε ότι έχουμε το μονοδιάστατο πίνακα A[8]:

46	55	12	42	94	18	06	67
----	----	----	----	----	----	----	----

τότε παρακάτω φαίνεται πως μετακινούνται τα στοιχεία με τον αλγόριθμο SelectionSort

Βήμα 1 (εύρεση του ελάχιστου των στοιχείων και ανταλλαγή με το πρώτο)

46	55	12	42	94	18	06	67
----	----	----	----	----	----	----	----

Βήμα 2 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το δεύτερο στοιχείο και κάτω)

06	55	12	42	94	18	46	67
----	----	----	----	----	----	----	----

Βήμα 3 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το τρίτο στοιχείο και κάτω)

06	12	55	42	94	18	46	67
----	----	----	----	----	----	----	----

Βήμα 4 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το τέταρτο στοιχείο και κάτω)

06	12	18	42	94	55	46	67
----	----	----	----	----	----	----	----

Βήμα 5 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το πέμπτο στοιχείο και κάτω)

06	12	18	42	94	55	46	67
----	----	----	----	----	----	----	----

Βήμα 6 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το έκτο στοιχείο και κάτω)

06	12	18	42	46	55	94	67
----	----	----	----	----	----	----	----

Βήμα 7 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το έβδομο στοιχείο και κάτω)

06	12	18	42	46	55	94	67
----	----	----	----	----	----	----	----

Τελική μορφή ταξινομημένου πίνακα (δεν χρειάζεται 8^η επανάληψη σύγκρισης, αφού όταν απομένουν δύο μόνο κελιά και στο πρώτο θέσεις τον μικρότερο αριθμό, τότε στο δεύτερο αναγκαστικά τίθεται ο μεγαλύτερος)

06	12	18	42	46	55	67	94
----	----	----	----	----	----	----	----

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΘΕΩΡΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΣΕ ΣΤΟΙΒΑ ΚΑΙ ΟΥΡΑ

- 1) Σε μια στοίβα 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
- i) Να προσδιορίσετε την τιμή του δείκτη **top** της παραπάνω στοίβας και να την σχεδιάσετε.
- ii) Αν εφαρμόσουμε τις παρακάτω λειτουργίες: **Απόθεση**, **Απόθεση**, **Απόθεση**, **Ώθηση X**, **Ώθηση Δ** και **Απόθεση** ποιά είναι η νέα τιμή της **top** και ποιά η τελική μορφή της στοίβας;
- 2) Η παραπάνω άσκηση να υλοποιηθεί με ουρά χρησιμοποιώντας, όπου Απόθεση **Εξαγωγή** και όπου Ώθηση **Εισαγωγή**. Επίσης αντί της top να δοθούν οι τιμές των δεικτών rear και front.
- 3) Σε μια άδεια στοίβα 10 θέσεων ωθούμε τα στοιχεία Ο, Σ, Ε, Τ, Λ. Με ποιά τρόπο πρέπει να ωθηθούν και να απωθηθούν τα δεδομένα ώστε η στοίβα να περιέχει τα δεδομένα Τ, Ε, Λ, Ο, Σ (σε αύξουσες θέσεις του πίνακα).
- 4) Η παραπάνω άσκηση να υλοποιηθεί με ουρά, χρησιμοποιώντας όπου Απόθεση **Εξαγωγή** και όπου Ώθηση **Εισαγωγή**. Επίσης αντί της top να δοθούν οι τιμές των δεικτών rear και front.

ΑΠΑΝΤΗΣΕΙΣ

- 1)
i)
- 10
9
8
7
6
5
4
3
2
1

Δ
Α
Μ
Γ
Σ

top →
- top=5
- 10
9
8
7
6
5
4
3
2
1

Χ
Γ
Σ

top →
- top=3
- ii) Η νέα τιμή της top είναι 3 και η στοίβα γίνεται:

2)

- i) Η αρχική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
Σ	Γ	Μ	Α	Δ					
front				rear					

οι τιμές της front και της rear είναι: front=1 και rear=5

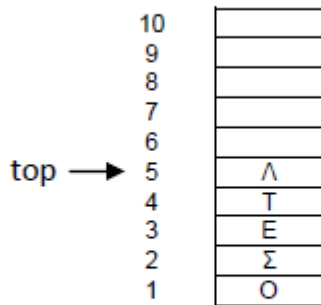
- ii) Η τελική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
				Δ	Χ	Δ			
front				rear					

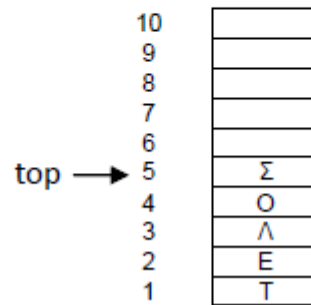
και οι τιμές της front και της rear γίνονται: front=5 και rear=7

3)

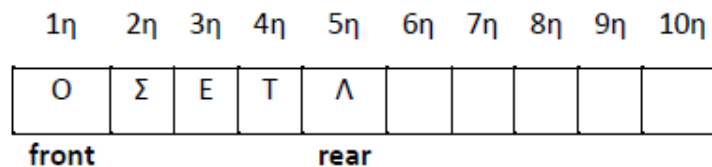
Η αρχική μορφή της στοίβας είναι:



Εκτελώντας τις λειτουργίες : Απώθηση, Απώθηση, Απώθηση, Απώθηση, Ύψωση Τ, Ύψωση Ε, Ύψωση Λ, Ύψωση Ο, Ύψωση Σ, τότε η τελική μορφή της στοίβας γίνεται:

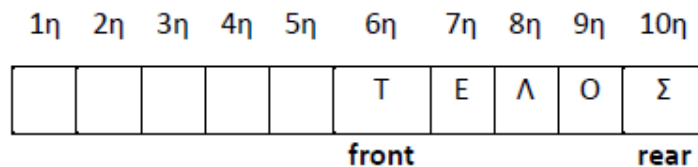


4) i) Η αρχική μορφή της ουράς είναι:



οι τιμές της front και της rear είναι: front=1 και rear=5

ii) Εκτελώντας τις λειτουργίες : Εξαγωγή, Εξαγωγή, Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή Τ, Εισαγωγή Ε, Εισαγωγή Λ, Εισαγωγή Ο, Εισαγωγή Σ, τότε η τελική μορφή της ουράς γίνεται



και οι τιμές της front και της rear γίνονται: front=6 και rear=10.

Υλοποίηση σε γλώσσα στοίβας με χρήση πίνακα.

Ένα οχηματαγωγό πλοίο, χωρητικότητας 250 αυτοκινήτων, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Τα οχήματα που επιβιβάζονται πρώτα είναι αυτά που θα αποβιβαστούν τελευταία. Στο λιμάνι του Πειραιά προσέρχονται τα αυτοκίνητα για αναχώρηση. Να γίνει πρόγραμμα το οποίο:

1. Να υπάρχει μενού επιλογής:

1. Επιβίβαση 2. Αποβίβαση 3. Έξοδος

2. Στη περίπτωση που επιλεγεί η Επιβίβαση θα διαβάζει τον αριθμό κυκλοφορίας καθενός από τα αυτοκίνητα που προσέρχονται και ο αριθμός κυκλοφορίας του να καταχωρείται στη στοίβα ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα αυτοκίνητο να τυπώνεται το ερώτημα "Υπάρχει άλλο αυτοκίνητο (Ν/Ο);". Αν ο χρήστης απαντήσει Ν (=ΝΑΙ), επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει Ο (=ΟΧΙ), σταματά η διαδικασία επιβίβασης και επιστρέφει το πρόγραμμα στο μενού Επιλογής.

3. Αν το πλοίο γεμίσει η επιβίβαση σταματά εμφανίζεται κατάλληλο μήνυμα και επιστρέφει το πρόγραμμα στο μενού επιλογής.

4. Στη περίπτωση που επιλεγεί η Αποβίβαση, εξάγει και εμφανίζει από την στοίβα ΟΧΗΜΑΤΑ όλους τους αριθμούς αυτοκινήτων που είχαν επιβιβαστεί στον ΠΕΙΡΑΙΑ, με τη σειρά που αποβιβάζονται. Στο τέλος να τυπώνεται το πλήθος των αυτοκινήτων που αποβιβάστηκαν στο λιμάνι της ΑΙΓΙΝΑΣ

Απάντηση

ΠΡΟΓΡΑΜΜΑ Λιμάνι

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ

ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, ΟΧΗΜΑΤΑ[250]

ΑΡΧΗ

τοπ ← 0

→ **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ**

ΓΡΑΨΕ ' Μενού Επιλογών'

ΓΡΑΨΕ ' 1. Επιβίβαση'

ΓΡΑΨΕ ' 2. Αποβίβαση'

ΓΡΑΨΕ ' 3. Έξοδος'

ΓΡΑΨΕ ' Δώσε επιλογή:'

ΔΙΑΒΑΣΕ επ1

→ **ΑΝ** επ1 = 1 **ΤΟΤΕ**

→ **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ**

ΓΡΑΨΕ ' Υπάρχει άλλο αυτοκίνητο (N/O);'

ΔΙΑΒΑΣΕ επ2

→ **ΑΝ** επ2 = 'N' **ΤΟΤΕ**

ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του αυτοκινήτου:'

ΔΙΑΒΑΣΕ αρ

τοπ ← τοπ + 1

ΟΧΗΜΑΤΑ[τοπ] ← αρ

ΑΝ τοπ = 250 **ΤΟΤΕ**

ΓΡΑΨΕ 'Το πλοίο γέμισε και δεν χωρά άλλα αμάξια'

ΤΕΛΟΣ_ΑΝ

→ **ΤΕΛΟΣ_ΑΝ**

→ **ΜΕΧΡΙΣ_ΟΤΟΥ** τοπ = 250 **Ή** επ2 = 'O'

→ **ΑΛΛΙΩΣ_ΑΝ** επ1 = 2 **ΤΟΤΕ**

πλ ← 0

ΟΣΟ τοπ >= 1 **ΕΠΑΝΑΛΑΒΕ**

ΓΡΑΨΕ 'Αποβιβάζεται το αυτοκίνητο με αριθμό κυκλοφορίας:', ΟΧΗΜΑΤΑ[τοπ]

ΟΧΗΜΑΤΑ[τοπ] ← ''

τοπ ← τοπ - 1

πλ ← πλ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Πλήθος οχημάτων που αποβιβάστηκαν στο λιμάνι της ΑΙΓΙΝΑΣ:', πλ

→ **ΤΕΛΟΣ_ΑΝ**

→ **ΜΕΧΡΙΣ_ΟΤΟΥ** επ1 = 3

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Λιμάνι

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΕΙΣΑΓΩΓΗ

- **Χαρακτηριστικά μονοδιάστατων πινάκων.** Είναι τα εξής:
1) Όνομα πίνακα: Ακολουθεί τους ίδιους κανόνες ονοματολογίας με τις μεταβλητές.
2) Μέγεθος πίνακα: Είναι ένας ακέραιος θετικός αριθμός που παραμένει **σταθερός** κατά τη διάρκεια εκτέλεσης του προγράμματος ή του αλγορίθμου και προσδιορίζει το πλήθος των θέσεων του πίνακα.
3) Τύπος δεδομένων του πίνακα: Μπορεί να είναι ΑΚΕΡΑΙΟΣ, ΠΡΑΓΜΑΤΙΚΟΣ, ΧΑΡΑΚΤΗΡΑΣ ή ΛΟΓΙΚΟΣ. Όλα τα στοιχεία του πρέπει να είναι του ίδιου τύπου.
- **Πως μπορούμε να δώσουμε τιμές σε έναν πίνακα.**
 Οι τιμές σε έναν πίνακα πρέπει να καταχωρούνται σε κάθε θέση ξεχωριστά και όχι σε ολόκληρο τον πίνακα. Αυτό μπορεί να γίνει όπως και με τις μεταβλητές με δύο τρόπους, είτε με την εντολή ΔΙΑΒΑΣΕ, είτε με την εντολή εκχώρησης τιμής.

Π.Χ. Έστω ότι έχουμε έναν πίνακα $A[4]$ ακεραίων.

$A[4] \leftarrow 15$	! Η τιμή 15 καταχωρείται στη θέση 4 του πίνακα.
$x \leftarrow 1$	! Στη μεταβλητή x καταχωρείται η τιμή 1.
$A[x] \leftarrow 4$	! Η τιμή 4 καταχωρείται στη θέση που δείχνει το x (δηλ. θέση 1).
$A[x+1] \leftarrow A[x]+1$	! Στη θέση 2 του πίνακα ($x+1=2$) καταχωρείται το περιεχόμενο της θέσης 1 αυξημένο κατά 1.
$A[3] \leftarrow A[x]+A[x+1]$	! Στη θέση 3 του πίνακα καταχωρείται το άθροισμα του 1 ^{ου} ($x=1$) και 2 ^{ου} ($x+1=2$) στοιχείου του πίνακα
ΔΙΑΒΑΣΕ $A[x+2]$	! Ζητείται μία τιμή από το χρήστη που αποθηκεύεται στη θέση 3 του πίνακα ($x+2=3$)
$A[2] \leftarrow A[2]+1$	! Το περιεχόμενο της 2 ^{ης} θέσης του πίνακα αυξάνεται κατά 1.

Παρατήρηση: Ο δείκτης ενός πίνακα πρέπει να είναι ή ακέραια σταθερά ή ακέραια μεταβλητή ή μία μαθηματική έκφραση στην οποία θα έχουν πάρει τιμές οι μεταβλητές που περιέχει και θα πρέπει να δίνει αποτέλεσμα θετικό ακέραιο μεγαλύτερο ή ίσο της μονάδας και μικρότερο ή ίσο του μεγέθους του πίνακα.

- **Το περιεχόμενο των θέσεων ενός πίνακα** περιέχει αρχικά απροσδιόριστες τιμές. Πρέπει λοιπόν πριν προχωρήσουμε στην επεξεργασία ενός πίνακα θα πρέπει είτε μέσω της εντολής ΔΙΑΒΑΣΕ ή με εντολές εκχώρησης τιμής να έχουν πάρει τιμή όλα τα στοιχεία του. Η επεξεργασία των στοιχείων του πίνακα γίνεται για κάθε στοιχείο ξεχωριστά και όχι ολικά για ολόκληρο τον πίνακα. Συνήθως για την επεξεργασία ενός πίνακα με N θέσεις χρησιμοποιούμε την δομή επανάληψης ΓΙΑ:

Για i από μέχρι N

Επεξεργασία του $A[i]$

Τέλος επανάληψης

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

1. Τι τιμές θα έχουν τα στοιχεία του πίνακα μετά την εκτέλεση των παρακάτω εντολών;

```

B[4] ← 0
X ← 1
B[X] ← B[4] + 6
B[X+1] ← B[X] + 2
B[X+2] ← B[X+1]
B[5] ← B[4] + 2
B[5] ← B[5] + X
    
```

ΛΥΣΗ

ΕΝΤΟΛΕΣ	ΠΙΝΑΚΑΣ Β					ΜΕΤΑΒΛΗΤΗ X
Αρχικές τιμές	1	2	3	4	5	?
	?	?	?	?	?	
B[4]←0	1	2	3	4	5	?
	?	?	?	0	?	
X←1	1	2	3	4	5	1
	?	?	?	0	?	
B[X]←B[4]+6	1	2	3	4	5	1
	6	?	?	0	?	
B[X+1]←B[X]+2	1	2	3	4	5	1
	6	8	?	0	?	

B[X+2]←B[X+1]	1 6	2 8	3 8	4 0	5 ?	1
B[5]←B[4]+2	1 6	2 8	3 8	4 0	5 2	1
B[5]←B[5]+X	1 6	2 8	3 8	4 0	5 3	1

Σημ. Το ? σημαίνει απροσδιόριστη τιμή.

2. Να γραφεί αλγόριθμος που θα διαβάσει 100 άρτιους αριθμούς και θα τους καταχωρεί σε πίνακα 100 θέσεων, κάνοντας τον κατάλληλο έλεγχο εγκυρότητας. Στη συνέχεια θα εμφανίζει τον μέσο όρο των στοιχείων του πίνακα που είναι μεγαλύτερα από 200.

ΛΥΣΗ

Λέγοντας έλεγχο εγκυρότητας των δεδομένων εισόδου εννοούμε ότι με τις κατάλληλες εντολές θα αποτρέπεται ο χρήστης από το να εισάγει τιμές που δεν θέλουμε. Αυτό συνήθως επιτυγχάνεται με μία επαναληπτική δομή της εξής μορφής:

ΑΡΧΗ_ΑΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ δεδομένο

ΜΕΧΡΙΣ_ΟΤΟΥ συνθήκη, όπου η συνθήκη θα εξασφαλίζει ότι ο χρήστης θα εισάγει την αποδεκτή τιμή. Στο συγκεκριμένο παράδειγμα θέλουμε η τιμή που εισάγει ο χρήστης να είναι άρτιος αριθμός οπότε η επαναληπτική δομή θα είναι:

ΑΡΧΗ_ΑΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ A[i]

ΜΕΧΡΙΣ_ΟΤΟΥ A[i] mod 2=0

Εναλλακτικά χρησιμοποιώντας την ΟΣΟ...ΕΠΑΝΑΛΑΒΕ μπορούμε να γράψουμε:

ΔΙΑΒΑΣΕ A[i]

ΟΣΟ A[i] mod 2 <>0 **ΕΠΑΝΑΛΑΒΕ**

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΓΟΡΙΘΜΟΣ Μέσος_Όρος

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΑΡΧΗ_ΑΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ A[i]

ΜΕΧΡΙΣ_ΟΤΟΥ A[i] mod 2=0

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Διάβασμα του πίνακα με έλεγχο εγκυρότητας

Σ← 0

ΠΛΗΘΟΣ←0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΑΝ A[i]>200 **ΤΟΤΕ**

Σ←Σ+A[i]

ΠΛΗΘΟΣ←ΠΛΗΘΟΣ+1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Υπολογισμός αθροίσματος και πλήθος στοιχείων που είναι μεγαλύτερα του 200

!Δεν βιαζόμαστε να γράψουμε ΜΟ←Σ/ΠΛΗΘΟΣ γιατί μπορεί το πλήθος να είναι 0

ΑΝ ΠΛΗΘΟΣ>0 **ΤΟΤΕ**

ΜΟ←Σ/ΠΛΗΘΟΣ

ΕΜΦΑΝΙΣΕ ΜΟ

ΑΛΛΙΩΣ

ΕΜΦΑΝΙΣΕ 'Δεν υπάρχουν αριθμοί μεγαλύτεροι του 200'

ΤΕΛΟΣ_ΑΝ

Υπολογισμός και εμφάνιση μέσου όρου

ΤΕΛΟΣ Μέσος_Όρος

3. Να γραφεί πρόγραμμα που θα ζητά από το χρήστη το πλήθος αριθμών που ο ίδιος θα δώσει και στη συνέχεια θα διαβάζει τους αριθμούς αυτούς και θα τους καταχωρεί σε έναν πίνακα ακεραίων. Το πλήθος των αριθμών που θα εισάγει ο χρήστης πρέπει να είναι μεταξύ 1 και 100. Στη συνέχεια θα εμφανίζει τους άρτιους αριθμούς του πίνακα που βρίσκονται στις περιττές θέσεις.

ΛΥΣΗ

Το πρόβλημα που παρουσιάζεται εδώ είναι ότι δεν είναι γνωστό εξ αρχής το μέγεθος του πίνακα. Δεν μπορούμε να δημιουργήσουμε έναν πίνακα αν δεν ξέρουμε το μέγεθος του. Αυτό που γνωρίζουμε είναι ότι είναι ένας αριθμός μεταξύ 1 και 100. Μπορούμε λοιπόν να χρησιμοποιήσουμε έναν πίνακα 100 θέσεων και αν ο χρήστης εισάγει για παράδειγμα 60 αριθμούς, να χρησιμοποιήσουμε τις 60 πρώτες θέσεις του πίνακα. Από και πέρα η επεξεργασία θα αφορά μόνο αυτές τις θέσεις.

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], πλήθος, i

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ πλήθος

ΜΕΧΡΙΣ_ΟΤΟΥ πλήθος > 0 **ΚΑΙ** πλήθος ≤ 100

} διάβασμα του πλήθους του πίνακα με έλεγχο εγκυρότητας

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** πλήθος

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} διάβασμα των θέσεων 1 έως πλήθος του πίνακα

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** πλήθος

ΑΝ i mod 2 = 1 **ΚΑΙ** A[i] mod 2 = 0 **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ A[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} αν η θέση του πίνακα είναι περιττός αριθμός και το περιεχόμενο της θέσης άρτιος, εμφάνισε το περιεχόμενο της θέσης αυτής

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

4. Δίνεται το παρακάτω τμήμα αλγορίθμου με αριθμημένες εντολές για εύκολη αναφορά σε αυτές. Κάθε εντολή περιέχει ένα ή δύο κενά (σημειωμένα με ...), που το καθένα αντιστοιχεί σε μία σταθερά ή μία μεταβλητή ή έναν τελεστή. Επίσης δίνεται πίνακας όπου κάθε γραμμή αντιστοιχεί στη διπλανή εντολή του τμήματος αλγορίθμου και κάθε στήλη σε μία θέση μνήμης (μεταβλητή). Η κάθε γραμμή του πίνακα παρουσιάζει το αποτέλεσμα που έχει η εκτέλεση της αντίστοιχης εντολής στη μνήμη: συγκεκριμένα, δείχνει την τιμή της μεταβλητής την οποία επηρεάζει η εντολή.

	Εντολές	Μνήμη								
		A	B	Γ	Δ	E	Z	X[1]	X[2]	X[3]
1.	A ← ...	4								
2.	Δ ← A + ...				7					
3.	Αν A ... Δ τότε Γ ← A αλλιώς Γ ← Δ Τέλος_αν			7						
4.	B ← ... - 1		3							
5.	E ← ... - ...					-1				
6.	... ← Δ + ...				6					
7.	Γ ← Γ ... E			8						
8.	Z ← ... - 1						2			
9.	X[...] ← Γ								8	
10.	X[Z ... 1] ← Δ							6		
11.	X[Z ... 1] ← X[Z] ... 1									7

Να γράψετε στο τετράδιό σας τον αριθμό της καθεμιάς εντολής και δίπλα να σημειώσετε τη σταθερά, τη μεταβλητή, ή τον τελεστή που πρέπει να αντικαταστήσει το κάθε κενό της εντολής ώστε να έχει το αποτέλεσμα που δίνεται στον πίνακα, ως εξής:

A. Για τις εντολές 1 και 2, να σημειώσετε σταθερές τιμές.

B. Για τις εντολές 3,7,10 και 11, να σημειώσετε τελεστές, και για τις υπόλοιπες, να σημειώσετε μεταβλητές. (Εξετάσεις 2007)

5. Δίνεται πίνακας A, 10 θέσεων ο οποίος στις θέσεις 1 έως 10 περιέχει αντίστοιχα τους αριθμούς 15,3,0,5,16,2,17,8,19,1, καθώς και τμήμα αλγορίθμου:

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 9 ΜΕ_ΒΗΜΑ 2

$K \leftarrow ((i+10) \bmod 10) + 1$

$A[i] \leftarrow A[K]$

ΕΚΤΥΠΩΣΕ i,K,A[i],A[K]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Ποιες τιμές τυπώνονται με την εντολή Εκτύπωσε i,K,A[i],A[K] καθώς εκτελείται το παραπάνω τμήμα αλγορίθμου;

(Εξετάσεις 2002)

6. Δίνεται ο ακόλουθος μονοδιάστατος πίνακας Τα με 8 στοιχεία που περιέχει τις ακόλουθες τιμές:

	1	2	3	4	5	6	7	8
T	1	1	2	1	4	1	1	4

καθώς και ο πίνακας Γ, 5 θέσεων με όλα του τα στοιχεία μηδενικά δηλαδή:

	1	2	3	4	5
Γ	0	0	0	0	0

Να γράψετε τις τιμές που θα περιέχει ο Γ, αν εκτελεστεί το ακόλουθο τμήμα αλγορίθμου:

ΓΙΑ θ ΑΠΟ 1 ΜΕΧΡΙ 8

$\Gamma[T[\theta]] \leftarrow \Gamma[T[\theta]] + 1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

7. Να γράψετε πρόγραμμα που θα διαβάζει 30 τριψήφιους θετικούς ακέραιους και θα τους καταχωρεί σε πίνακα 30 θέσεων. Το πρόγραμμα θα εμφανίζει τον μέσο όρο αυτών που αποθηκεύτηκαν σε περιττές θέσεις και τουλάχιστον ένα από τα ψηφία τους είναι 8. Θεωρήστε ότι ο χρήστης θα δώσει έγκυρες τιμές.

8. Να γράψετε αλγόριθμο που θα διαβάζει δύο πίνακες 10 θέσεων και θα δημιουργεί έναν πίνακα 20 θέσεων που στις 10 πρώτες θέσεις θα περιέχει τα στοιχεία του πρώτου πίνακα και στις 10 επόμενες τα στοιχεία του δεύτερου πίνακα.

9. Να γράψετε αλγόριθμο ο οποίος για δεδομένο πίνακα N θέσεων θα διαβάζει έναν ακέραιο αριθμό που θα ανήκει στο κλειστό διάστημα $[1,N]$ και δημιουργεί έναν καινούριο πίνακα N-1 θέσεων που θα περιέχει τα στοιχεία του πρώτου πίνακα εκτός του στοιχείου που η θέση του καθορίζεται από τον ακέραιο αριθμό που έδωσε ο χρήστης.

ΑΠΛΕΣ ΕΦΑΡΜΟΓΕΣ ΜΟΝΟΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ **ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ**

10. Να γραφεί αλγόριθμος που θα διαβάζει τη θερμοκρασία 50 πόλεων που σημειώθηκε στις 12 το μεσημέρι μιας ημέρας και θα εμφανίζει το πλήθος των πόλεων που είχαν θερμοκρασία μεγαλύτερη από τη μέση θερμοκρασία όλων των πόλεων.

ΛΥΣΗ

Καταρχήν θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα 50 θέσεων, που σε κάθε θέση του θα αποθηκεύσουμε τη θερμοκρασία της αντίστοιχης πόλης:

	1	2	3		50
Θ					

Στη συνέχεια θα υπολογίσουμε τον μέσο όρο των τιμών του πίνακα και θα προσπελάσουμε όλα τα στοιχεία του πίνακα από το 1^ο έως το 50^ο, αυξάνοντας το περιεχόμενο μίας μεταβλητής κατά 1 κάθε φορά που η τιμή του πίνακα που προσπελάζουμε είναι μεγαλύτερη από τον μέσο όρο που υπολογίσαμε πριν.

Ακολουθεί ο αλγόριθμος:

ΑΓΟΡΙΘΜΟΣ Άσκηση

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    ΔΙΑΒΑΣΕ Θ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

} διάβασμα του πίνακα

Σ ← 0

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    Σ ← Σ + Θ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ ← Σ / 50

```

} (για κάθε τιμή του πίνακα από την 1^η έως την 50^η πρόσθεσε στο Σ την αντίστοιχη τιμή)

πλήθος ← 0

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    ΑΝ Θ[i] > ΜΟ ΤΟΤΕ
        πλήθος ← πλήθος + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

} (για κάθε τιμή του πίνακα από την 1^η μέχρι την 50^η, αν η τιμή είναι μεγαλύτερη του ΜΟ, αύξησε το πλήθος κατά 1)

ΕΜΦΑΝΙΣΕ πλήθος

ΤΕΛΟΣ Άσκηση

11. Να γραφεί αλγόριθμος ο οποίος θα διαβάζει το όνομα και το βαθμό πτυχίου 100 φοιτητών και θα καταχωρεί τα στοιχεία αυτά σε δύο πίνακες. Στη συνέχεια να εμφανίζει τα ονόματα των φοιτητών με βαθμό πτυχίου μεγαλύτερου του 7.

ΛΥΣΗ

Θα χρησιμοποιήσουμε δύο πίνακες 100 θέσεων που στον πρώτο θα αποθηκεύσουμε το όνομα και στον δεύτερο το βαθμό του αντίστοιχου φοιτητή. Οι πίνακες αυτοί λέγονται παράλληλοι αφού για παράδειγμα στην 3^η θέση του πρώτου πίνακα θα είναι καταχωρημένο το όνομα του 3^{ου} φοιτητή και στην 3^η θέση του δεύτερου πίνακα ο βαθμός του φοιτητή αυτού. Οι πίνακες που θα χρησιμοποιήσουμε θα έχουν την παρακάτω μορφή:

	ΟΝΟΜΑ	ΒΑΘΜΟΣ
1		
2		
.....		
100		

Ο αλγόριθμος είναι ο εξής:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΕΜΦΑΝΙΣΕ 'Δώσε το όνομα και το βαθμό του', i, 'ου φοιτητή'

ΔΙΑΒΑΣΕ ΟΝΟΜΑ[i], ΒΑΘΜΟΣ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΑΝ ΒΑΘΜΟΣ[i] > 7 **ΤΟΤΕ** **ΕΜΦΑΝΙΣΕ** ΟΝΟΜΑ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ Άσκηση

12. Μια εταιρεία αποφάσισε να κρατήσει κάποια στατιστικά στοιχεία για τους 40 εργαζομένους της. Αποφάσισε λοιπόν να κρατάει για κάθε εργαζόμενο το όνομά του, το φύλο του, που πρέπει να είναι 'Α' για τους άνδρες και 'Γ' για τις γυναίκες και το μηνιαίο μισθό του. Να γραφεί πρόγραμμα το οποίο:

α) Θα διαβάζει τα παραπάνω στοιχεία για τους εργαζόμενους κάνοντας παράλληλα έλεγχο εγκυρότητας για το φύλο ώστε να είναι 'Α' ή 'Γ'.

β) Θα υπολογίζει το μέσο μισθό των ανδρών της εταιρείας.

γ) Θα εμφανίζει το όνομα των γυναικών της εταιρείας που έχουν μισθό μικρότερο από το μέσο μισθό των ανδρών.

ΛΥΣΗ

Θα χρησιμοποιήσουμε τρεις παράλληλους πίνακες 40 θέσεων που στον πρώτο θα καταχωρίσουμε το όνομα, στον δεύτερο το φύλο και στον τρίτο το μηνιαίο μισθό κάθε εργαζόμενου:

	ΟΝ	Φ	Μ
1			
2			
3			
.....			
40			

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[40], Φ[40]

ΠΡΑΓΜΑΤΙΚΕΣ: Μ[40], Σ, ΜΟ

ΑΚΕΡΑΙΕΣ: i, ΠΛ_ΑΝΔ

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 40

ΔΙΑΒΑΣΕ ΟΝ[i], Μ[i]

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Φ[i]

ΜΕΧΡΙΣ_ΟΤΟΥ Φ[i]='Α' **Η'** Φ[i]='Τ'

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

(α ερώτημα): διάβασμα των πινάκων ΟΝ, Μ και του Φ με έλεγχο εγκυρότητας

Σ←0

ΠΛ_ΑΝΔ←0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 40

ΑΝ Φ[i]='Α' **ΤΟΤΕ**

Σ←Σ+Μ[i]

ΠΛ_ΑΝΔ←ΠΛ_ΑΝΔ+1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

(β ερώτημα): υπολογισμός αθροίσματος μισθών και πλήθους των ανδρών για υπολογισμό του μέσου μισθού τους

ΜΟ←Σ/ΠΛ_ΑΝΔ

! Το πλήθος ανδρών μπορεί να είναι μηδενικό.

ΑΝ ΠΛ_ΑΝΔ<>0 **ΤΟΤΕ**

ΜΟ←Σ/ΠΛ_ΑΝΔ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 40

ΑΝ Φ[i]='Τ' **ΚΑΙ** Μ[i]<ΜΟ **ΤΟΤΕ**

ΓΡΑΨΕ ΟΝ[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δεν υπάρχουν άνδρες στην εταιρεία'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

(γ ερώτημα): εφόσον το πλήθος των ανδρών δεν είναι μηδενικό, υπολογίζουμε το μέσο μισθό τους. Μετά κάνουμε προσπέλαση στους πίνακες και αν βρούμε γυναίκα με μισθό μικρότερο του ΜΟ, εμφανίζουμε το όνομά της

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

13. Μια εταιρεία δημοσκοπήσεων θέτει σ' ένα δείγμα 2.000 πολιτών ένα ερώτημα. Για την επεξεργασία των δεδομένων να αναπτύξετε αλγόριθμο που:

1. θα διαβάζει το φύλο του πολίτη (Α = Άνδρας, Γ = Γυναίκα) και θα ελέγχει την ορθή εισαγωγή,
2. θα διαβάζει την απάντηση στο ερώτημα, η οποία μπορεί να είναι «ΝΑΙ», «ΟΧΙ», «ΔΕΝ ΞΕΡΩ» και θα ελέγχει την ορθή εισαγωγή,
3. να υπολογίζει και να εμφανίζει το πλήθος των ατόμων που απάντησαν «ΝΑΙ»,
4. στο σύνολο των ατόμων που απάντησαν «ΝΑΙ» θα υπολογίζει και θα εμφανίζει το ποσοστό των ανδρών και το ποσοστό των γυναικών. (Εξετάσεις 2004)

14. Ένας μοναχικός βοσκός την τελευταία χρονιά είχε 30 πρόβατα και 30 αγελάδες. Και από τα δύο είδη ζώων παράγει γάλα το οποίο το πουλάει για 0.5 € το λίτρο το πρόβειο και 0.4 € το λίτρο το αγελαδινό. Ο μοναχικός βοσκός, για να ξεχωρίζει τα πρόβατα και τις αγελάδες του, αποφάσισε να δώσει σε κάθε ζώο του ένα όνομα. Να κάνετε πρόγραμμα το οποίο να διαβάζει για τα 60 ζώα του βοσκού το χαρακτηριστικό γράμμα «Π» για πρόβατο και «Α» για αγελάδα. Στη συνέχεια, αν πρόκειται για πρόβατο, θα καταχωρίζει το όνομα του και την ποσότητα γάλατος σε δύο πίνακες, ΟνΠ και ΠοσΠ αντίστοιχα, ενώ, αν πρόκειται για αγελάδα, σε δύο πίνακες, ΟνΑ και ΠοσΑ αντίστοιχα. Στη συνέχεια να εμφανίζει το όνομα των προβάτων που βγάζουν κέρδος περισσότερο από το μέσο όρο κέρδους των αγελάδων, καθώς και το όνομα των αγελάδων που βγάζουν κέρδος περισσότερο από τον μέσο όρο κέρδους των προβάτων.

Σημείωση: Θεωρήστε ότι κατά την εισαγωγή των τιμών ο βοσκός θα δώσει τα στοιχεία των αγελάδων και των προβάτων, μπερδεμένα, δηλαδή μπορεί να δώσει πρώτα τα στοιχεία μιας αγελάδας και μετά τα στοιχεία ενός πρόβατου, δίνοντας όμως σωστό αριθμό προβάτων και αγελάδων. Επιπλέον, δεν χρειάζεται να γίνει έλεγχος εγκυρότητας των τιμών.

15. Σε έναν διαγωνισμό Μαθηματικών θα λάβουν μέρος άτομα τα οποία στις πανελλήνιες εξετάσεις έχουν πάρει βαθμό μεγαλύτερο ή ίσο του 90. Στον διαγωνισμό αυτό επίσης, επειδή οι συμμετοχές θα είναι πάρα πολλές, αποφασίστηκε να συμμετάσχουν μόνο οι πρώτοι 200 που έχουν κάνει αίτηση και συγχρόνως θα ικανοποιούν τον παραπάνω βαθμολογικό περιορισμό. Να γράψετε αλγόριθμο ο οποίος θα διαβάσει το όνομα και τον βαθμό στα Μαθηματικά ενός υποψήφιου και θα καταχωρίζει σε έναν πίνακα το όνομα του, εφόσον ο βαθμός του στα Μαθηματικά είναι μεγαλύτερος από το 90. Αν ο βαθμός του δεν πληρεί αυτόν τον περιορισμό, τότε ο υποψήφιος θα αγνοείται. Ο αλγόριθμος θα διαβάσει ονόματα μέχρι να συμπληρωθούν και οι 200 θέσεις. Τέλος ο αλγόριθμος αυτός να εμφανίζει τα ονόματα των υποψηφίων που θα συμμετάσχουν στον διαγωνισμό.

ΜΕΓΙΣΤΟ-ΕΛΑΧΙΣΤΟ ΣΤΟΙΧΕΙΟ ΜΟΝΟΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

16. Να γίνει αλγόριθμος που θα δέχεται τα ονόματα 30 πόλεων της Ελλάδας καθώς και τον πληθυσμό τους. Ο αλγόριθμος θα εμφανίζει την πόλη με το μικρότερο πληθυσμό. Θεωρήστε ότι καμία πόλη δεν έχει τον ίδιο πληθυσμό με τις υπόλοιπες.

ΛΥΣΗ

Θα χρησιμοποιήσουμε τους εξής παράλληλους πίνακες 30 θέσεων:

	1	2	3		30
ΟΝ					

	1	2	3		30
ΠΛΗΘ					

Η άσκηση δεν μας ζητάει το ελάχιστο στοιχείο του πίνακα ΠΛΗΘ, αλλά το όνομα της πόλης με τον ελάχιστο πληθυσμό. Θα πρέπει λοιπόν να εντοπίσουμε τη θέση του πίνακα ΠΛΗΘ που βρίσκεται το ελάχιστο στοιχείο και να εμφανίσουμε το αντίστοιχο στοιχείο του πίνακα ΟΝ. Αν για παράδειγμα βρούμε ότι το ελάχιστο στοιχείο του πίνακα ΠΛΗΘ βρίσκεται στην 8^η θέση, θα πρέπει να εμφανίσουμε το 8^ο στοιχείο του πίνακα ΟΝ. Για τον υπολογισμό του ελάχιστου στοιχείου του πίνακα ΠΛΗΘ και της θέσης του, καταχωρούμε αρχικά στη μεταβλητή **min** το 1^ο στοιχείο του πίνακα, και στη μεταβλητή **θέση_min** το 1. Στη συνέχεια συγκρίνουμε ένα-ένα όλα τα στοιχεία του πίνακα (ξεκινώντας από το 2^ο) με το min και αν βρούμε κάποιο στοιχείο μικρότερο, καταχωρούμε στο min το στοιχείο αυτό και στη μεταβλητή θέση_min τη θέση που βρισκόμαστε. Ακολουθεί ο αλγόριθμος:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30

ΔΙΑΒΑΣΕ ΟΝ[i], ΠΛΗΘ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

min ← ΠΛΗΘ[1]

θέση_min ← 1

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 30

ΑΝ ΠΛΗΘ[i] < min **ΤΟΤΕ**

min ← ΠΛΗΘ[i]

θέση_min ← i

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΕΜΦΑΝΙΣΕ ΟΝ[θέση_min]

ΤΕΛΟΣ Άσκηση

Εναλλακτικά, για την εύρεση της πόλης με το μικρότερο πληθυσμό, μπορούμε να χρησιμοποιήσουμε το παρακάτω τμήμα αλγορίθμου:

min ← ΠΛΗΘ[1]

min_ON ← ΟΝ[1]

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 30

ΑΝ ΠΛΗΘ[i] < min **ΤΟΤΕ**

min ← ΠΛΗΘ[i]

min_ON ← ΟΝ[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΕΜΦΑΝΙΣΕ min_ON

17. Μια βιβλιοθήκη διαθέτει 4.000 τίτλους βιβλίων. Να γίνει πρόγραμμα που θα διαβάσει τους τίτλους αυτούς καθώς και τον αριθμό αντιτύπων τους και θα τους καταχωρεί σε δύο πίνακες T και AP_ANT αντίστοιχα. Το πρόγραμμα να εμφανίζει τους τίτλους των βιβλίων που έχουν το μέγιστο αριθμό αντιτύπων.

ΛΥΣΗ

Η άσκηση αυτή διαφέρει από την προηγούμενη στο γεγονός ότι το μέγιστο αριθμό αντιτύπων είναι δυνατόν να τον έχουν παραπάνω από ένα βιβλία (δεν διευκρινίζεται όπως στην προηγούμενη άσκηση ότι τα στοιχεία του πίνακα AP_ANT είναι διαφορετικά μεταξύ τους). Θα πρέπει λοιπόν αφού βρούμε το μέγιστο στοιχείο στον πίνακα AP_ANT, να ξαναπροσπελάσουμε τον πίνακα και αν βρίσκουμε στοιχείο που ισούται με το μέγιστο στοιχείο που βρήκαμε πριν, να εμφανίζουμε το αντίστοιχο στοιχείο του πίνακα T.

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: AP_ANT[4000],max,i

ΧΑΡΑΚΤΗΡΕΣ: T[4000]

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4000

ΔΙΑΒΑΣΕ T[i] , AP_ANT[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4000

ΑΝ i=1 ΤΟΤΕ

max←AP_ANT[i]

ΑΛΛΙΩΣ

ΑΝ AP_ANT[i]>max **ΤΟΤΕ**

max←AP_ANT[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4000

ΑΝ AP_ANT[i]=max **ΤΟΤΕ**

ΓΡΑΨΕ T[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

18. Να γράψετε αλγόριθμο ο οποίος θα διαβάσει το όνομα 100 ταινιών ενός Video Club καθώς και το πλήθος των ενοικιάσεων για κάθε ταινία τον τελευταίο μήνα. Να εμφανίζει:

- i. τον μέγιστο αριθμό ενοικιάσεων μιας ταινίας,
- ii. το όνομα της ταινίας με τον ελάχιστο αριθμό ενοικιάσεων,
- iii. τις ταινίες που είχαν τον μέγιστο αριθμό ενοικιάσεων.

19. Ένας περιπτεράς έχει καταγράψει σε κατάλληλο μονοδιάστατο πίνακα τις μηνιαίες εισπράξεις που πραγματοποίησε κατά το προηγούμενο έτος.

A. Να σχεδιάσετε αλγόριθμο, ο οποίος:

- α. να διαβάσει τα στοιχεία του πίνακα αυτού.
- β. να εμφανίζει τις συνολικές εισπράξεις ανά τρίμηνο.
- γ. να εμφανίζει τον αριθμό του διμήνου με τις περισσότερες εισπράξεις.

B. Αποφάσισε να συνεταιριστεί με κάποιον γνωστό του και να λειτουργούν το περίπτερο από κοινού ένα μήνα ο καθ' ένας εναλλάξ. Δηλαδή τους περιττούς μήνες θα το λειτουργεί και θα εισπράττει τα έσοδά του ο ένας και τους άρτιους μήνες ο άλλος ή αντίστροφα. Ο αλγόριθμος, αξιοποιώντας τα στοιχεία της περσινής χρονιάς, θα πρέπει να ελέγχει και να εμφανίζει μήνυμα σχετικά με το αν συμφέρει τον κύριο Αρβίλογλου να διατηρήσει τη λειτουργία του περιπτέρου τους άρτιους ή τους περιττούς μήνες ώστε να έχει τα περισσότερα έσοδα.

20. Μια νομαρχία διοργάνωσε το 2008 σεμινάριο εθελοντικής δασοφυτεύσεως, το οποίο παρακολούθησαν 500 άτομα. Η Πυροσβεστική Υπηρεσία ζήτησε στοιχεία σχετικά με την ηλικία, το φύλο και το μορφωτικό

επίπεδο εκπαίδευσης κάθε εθελοντή, προκειμένου να εξαγάγει στατιστικά στοιχεία. Να γράψετε αλγόριθμο, ο οποίος:

i. διαβάσει για κάθε άτομο

- το ονοματεπώνυμο,
- το έτος γέννησης (χωρίς να απαιτείται έλεγχος εγκυρότητας),
- το φύλο, με αποδεκτές τιμές το "Α" για τους άνδρες και το "Γ" για τις γυναίκες,
- το μορφωτικό επίπεδο εκπαίδευσης, με αποδεκτές τιμές "Π", "Δ" ή "Τ", που αντιστοιχούν σε Πρωτοβάθμια, Δευτεροβάθμια ή Τριτοβάθμια Εκπαίδευση, και τα καταχωρεί σε κατάλληλους μονοδιάστατους πίνακες.

ii. υπολογίζει και εμφανίζει το πλήθος των ατόμων με ηλικία μικρότερη των 30 ετών.

iii. υπολογίζει και εμφανίζει το ποσοστό των γυναικών με επίπεδο Τριτοβάθμιας Εκπαίδευσης στο σύνολο των εθελοντριών.

iv. εμφανίζει τα ονόματα των ατόμων με τη μεγαλύτερη ηλικία. (Εξετάσεις 2008)

ΠΡΟΒΛΗΜΑΤΑ ΑΠΟΦΑΣΗΣ ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

21. Έστω ένας πίνακας B, 30 θέσεων που περιέχει τους βαθμούς των 30 μαθητών μιας τάξης. Να γραφεί αλγόριθμος που θα εμφανίζει αν υπάρχουν μαθητές με βαθμό μικρότερο του 10.

ΛΥΣΗ

Στα προβλήματα αυτού του τύπου, ζητείται αν υπάρχουν στοιχεία/ο του πίνακα που ικανοποιούν μία συνθήκη. Στο συγκεκριμένο πρόβλημα ζητείται για το αν υπάρχει έστω και ένα στοιχείο του πίνακα με τιμή μικρότερη του 10. Χρησιμοποιούμε λοιπόν μία **λογική μεταβλητή** και την αρχικοποιούμε έστω με την τιμή **ψευδής**. Στη συνέχεια διατρέχουμε όλα τα στοιχεία του πίνακα και αν βρεθεί έστω και ένα στοιχείο που ικανοποιεί την συνθήκη που θέλουμε, καταχωρούμε στη μεταβλητή αυτή τιμή αντίθετη της αρχικής, δηλαδή αληθής. Μετά την προσπέλαση του πίνακα, η τιμή της λογικής μεταβλητής θα είναι και η απάντηση στο πρόβλημά μας, δηλαδή αν είναι ψευδής δεν θα έχει βρεθεί κανένα στοιχείο που ικανοποιεί την συνθήκη, ενώ αν είναι αληθής, θα έχει βρεθεί.

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΔΕΔΟΜΕΝΑ //B//

! Ο πίνακας B θεωρείται δεδομένος

υπάρχουν←ψευδής

!αρχικοποίηση της λογικής μεταβλητής

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30

ΑΝ B[i]<10 ΤΟΤΕ

υπάρχουν←αληθής

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} προσπέλαση του πίνακα και αν βρεθεί στοιχείο με τιμή μικρότερη του 10, αλλάζουμε την τιμή της λογικής μεταβλητής

ΑΝ υπάρχουν=ψευδής ΤΟΤΕ

ΕΜΦΑΝΙΣΕ 'δεν υπάρχουν μαθητές με βαθμό κάτω του 10'

ΑΛΛΙΩΣ

ΕΜΦΑΝΙΣΕ 'υπάρχουν μαθητές με βαθμό κάτω του 10'

ΤΕΛΟΣ_ΑΝ

} Έλεγχος της λογικής μεταβλητής

ΤΕΛΟΣ Άσκηση

Παρατήρηση: Δεν επιτρέπεται κατά τον έλεγχο των στοιχείων του πίνακα μέσα στην επανάληψη, η λογική μεταβλητή να πάρει ίδια τιμή με την αρχική. Το παραπάνω πρόβλημα μπορούμε να το λύσουμε και με χρήση μετρητή, που θα μετράει το πλήθος των μαθητών με βαθμό κάτω του 10. Μετά την προσπέλαση του πίνακα, ελέγχουμε την τιμή του μετρητή και αν αυτή είναι 0, τότε δεν θα υπάρχουν μαθητές με βαθμό κάτω του 10, αλλιώς θα υπάρχουν.

22. Να γίνει αλγόριθμος που για έναν πίνακα 100 θέσεων θα εμφανίζει μήνυμα για το αν όλα του τα στοιχεία είναι θετικοί αριθμοί ή μηδέν. Αν βρεί έστω και ένα στοιχείο αρνητικό ο αλγόριθμος θα εμφανίζει το μήνυμα και θα σταματά.

ΛΥΣΗ

Όπως στην προηγούμενη άσκηση, θα αρχικοποιήσουμε μία λογική μεταβλητή πριν αρχίσουμε την προσπέλαση των στοιχείων του πίνακα και αν βρούμε ένα στοιχείο του πίνακα μικρότερο του 0, θα αλλάξουμε την τιμή της λογικής μεταβλητής. Η διαφορά με την προηγούμενη άσκηση είναι ότι θέλουμε αν

βρεθεί ένα στοιχείο του πίνακα που ικανοποιεί τη συνθήκη, να σταματήσει η προσπέλαση των στοιχείων του πίνακα. Δεν ξέρουμε τον αριθμό των επαναλήψεων που πρέπει να γίνουν, αφού μπορεί ο πρώτος αρνητικός αριθμός να βρεθεί στην 1^η θέση, στην 65^η, ή στην 100^η. Άρα δεν μπορούμε να χρησιμοποιήσουμε τη δομή επανάληψης ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ. Θα χρησιμοποιήσουμε λοιπόν την ΟΣΟ...ΕΠΑΝΑΛΑΒΕ όπως φαίνεται στον αλγόριθμο:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΔΕΔΟΜΕΝΑ //Α//

υπάρχει←ψευδής

i←1

ΟΣΟ i<=100 **ΚΑΙ** υπάρχει=ψευδής **ΕΠΑΝΑΛΑΒΕ** !όσο δεν φτάσαμε στο τέλος του πίνακα και όσο δεν έχει βρεθεί αρνητικό στοιχείο, συνέχισε την επανάληψη.
ΑΝ A[i]<0 **ΤΟΤΕ**
 υπάρχει←αληθής

ΤΕΛΟΣ_ΑΝ

i ← i+1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ υπάρχει=ψευδής **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ 'όλα τα στοιχεία είναι θετικοί ή μηδέν'

ΑΛΛΙΩΣ

ΕΜΦΑΝΙΣΕ 'δεν είναι όλα τα στοιχεία θετικοί ή μηδέν'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ Άσκηση

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

23. Να γράψετε αλγόριθμο ο οποίος θα δέχεται έναν πίνακα 50 θέσεων και θα εμφανίζει μήνυμα για το αν τα στοιχεία του είναι συμμετρικά, δηλαδή το στοιχείο της 1ης θέσης είναι ίσο με το στοιχείο της 50ής θέσης, αυτό της 2ης ίσο με αυτό της 49ης κ.ο.κ.

24. Δίνεται πίνακας A[N] ακέραιων και θετικών αριθμών, καθώς και πίνακας B[N-1] πραγματικών και θετικών αριθμών. Να γράψετε αλγόριθμο ο οποίος να ελέγχει αν κάθε στοιχείο B[i] είναι ο μέσος όρος των στοιχείων A[i] και A[i + 1], δηλαδή αν $B[i] = (A[i] + A[i + 1]) / 2$. Σε περίπτωση που ισχύει, τότε να εμφανίζεται το μήνυμα «Ο πίνακας B είναι ο τρέχων μέσος του A», διαφορετικά να εμφανίζεται το μήνυμα «Ο πίνακας B δεν είναι ο τρέχων μέσος του A».

Για παράδειγμα:

Έστω ότι τα στοιχεία του πίνακα A είναι 1, 3, 5, 10, 15 και ότι τα στοιχεία του πίνακα B είναι 2, 4, 7.5, 12.5 .

Τότε ο αλγόριθμος θα εμφανίσει το μήνυμα «Ο πίνακας B είναι ο τρέχων μέσος του A», διότι:

$2 = (1+3)/2$, $4 = (3 + 5)/2$, $7.5 = (5 + 10)/2$, $12.5 = (10 + 15) / 2$

(Εξετάσεις 2005)

ΔΙΣΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ-ΕΙΣΑΓΩΓΗ

Έστω ότι μας τίθεται το εξής πρόβλημα:

Να γραφεί αλγόριθμος που θα διαβάζει τους βαθμούς 30 μαθητών μιας τάξης σε 14 μαθήματα.

Είναι προφανές ότι στο συγκεκριμένο πρόβλημα δεν μπορούμε να χρησιμοποιήσουμε έναν μοναδιάστατο πίνακα 14 θέσεων για τους βαθμούς των μαθημάτων ενός μαθητή, γιατί έτσι θα πρέπει να φτιάξουμε 30 τέτοιους πίνακες, έναν για κάθε μαθητή! Σε τέτοιες περιπτώσεις χρησιμοποιούμε στον προγραμματισμό πίνακες δύο διαστάσεων, δηλαδή πίνακες με γραμμές και στήλες. Για το παραπάνω παράδειγμα μπορούμε να χρησιμοποιήσουμε έναν πίνακα με όνομα π.χ. ΒΑΘ, με 30 γραμμές και 14 στήλες :

ΒΑΘ	1	2	3		14
1					
2					
3					
...	...	...	...	...	...
30					

Οι γραμμές του πίνακα αντιστοιχούν στους μαθητές και οι στήλες στα μαθήματα, δηλαδή το στοιχείο π.χ.

ΒΑΘ[8 , 5] του πίνακα περιέχει το βαθμό του 8^{ου} μαθητή στο 5^ο μάθημα. Αναφερόμαστε στα στοιχεία του πίνακα γράφοντας πρώτα το όνομά του και μέσα σε αγκύλες δύο δείκτες χωρισμένους με κόμμα , που ο

πρώτος αφορά την γραμμή και ο δεύτερος τη στήλη του πίνακα. Το πλήθος των στοιχείων του πίνακα είναι πλήθος γραμμών $\times$ πλήθος στηλών, δηλαδή ο παραπάνω πίνακας περιέχει $30 \times 14 = 420$ στοιχεία. Θα μπορούσαμε επίσης για το παραπάνω παράδειγμα να χρησιμοποιήσουμε έναν πίνακα 14×30 , δηλαδή 14 γραμμών και 30 στηλών, όπου οι γραμμές θα αντιστοιχούσαν στα μαθήματα και οι στήλες στους μαθητές:

ΒΑΘ	1	2	3		30
1					
2					
3					
...	...	...	...	...	...
14					

Η σχεδίαση που ακολουθούμε εξαρτάται από εμάς.

Οι παρατηρήσεις που κάναμε για τους μονοδιάστατους πίνακες ισχύουν και εδώ, δηλαδή:

- Τα χαρακτηριστικά ενός πίνακα δύο διαστάσεων είναι το **όνομά** του, το **μέγεθός** του και ο **τύπος δεδομένων** του πίνακα. Τα στοιχεία του πρέπει να είναι όλα του ίδιου τύπου.
π.χ. για να δηλώσουμε σε ένα πρόγραμμα έναν πίνακα ακεραίων με όνομα A και μέγεθος 20×15 γράφουμε στο τμήμα δηλώσεων του προγράμματος:
ΑΚΕΡΑΙΕΣ: A[20,15]
- Δίνουμε τιμές στις θέσεις ενός πίνακα είτε με εντολές εκχώρησης π.χ. $A[2,7] \leftarrow 12$, είτε με εντολές **ΔΙΑΒΑΣΕ, π.χ. ΔΙΑΒΑΣΕ A[5,9]**. Για να διαβάσουμε όλα τα στοιχεία ενός πίνακα διαστάσεων $N \times M$, χρησιμοποιούμε εμφωλευμένα ΓΙΑ:

ΔΙΑΒΑΣΜΑ ΚΑΤΑ ΓΡΑΜΜΕΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M

ΔΙΑΒΑΣΕ A[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΜΑ ΚΑΤΑ ΣΤΗΛΕΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N

ΔΙΑΒΑΣΕ A[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

- Το περιεχόμενο των θέσεων του πίνακα είναι αρχικά απροσδιόριστο, οπότε για να προχωρήσουμε στην επεξεργασία των στοιχείων του θα πρέπει να έχουν διαβαστεί ή να έχουν εκχωριστεί τιμές στα στοιχεία του, εκτός και αν από την εκφώνηση της άσκησης ο πίνακας είναι δεδομένος.

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

25. Να γραφούν οι κατάλληλες εντολές για την εμφάνιση των στοιχείων ενός πίνακα A 10×5 , κατά γραμμές και κατά στήλες.

ΛΥΣΗ

A	1	2	3	4	5
1	A[1,1]	A[1,2]	A[1,3]	A[1,4]	A[1,5]
2	A[2,1]	A[2,2]	A[2,3]	A[2,4]	A[2,5]
....					
10	A[10,1]	A[10,2]	A[10,3]	A[10,4]	A[10,5]

Τα στοιχεία της πρώτης γραμμής του πίνακα είναι τα A[1,1], A[1,2], A[1,3], A[1,4], A[1,5].

Παρατηρούμε ότι παραμένει σταθερός ο δείκτης της γραμμής=1 και μεταβάλλεται ο δείκτης της στήλης από 1 μέχρι 5. Για την εμφάνιση λοιπόν της πρώτης γραμμής μπορούμε να γράψουμε:

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5

ΕΜΦΑΝΙΣΕ A[1, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Ομοίως, για την εμφάνιση της 2^{ης} γραμμής:

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5

ΕΜΦΑΝΙΣΕ A[2, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Της 10^{ης} γραμμής:

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5

ΕΜΦΑΝΙΣΕ A[10, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Παρατηρούμε εδώ ότι επαναλαμβάνεται το ίδιο τμήμα εντολών από την 1^η έως την 10^η γραμμή. Μπορούμε λοιπόν αυτές τις εντολές να τις γράψουμε μέσα σε ένα ΓΙΑ , από 1 μέχρι 10 δηλαδή:

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5
    ΕΜΦΑΝΙΣΕ A[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Ομοίως βρίσκουμε ότι για την εμφάνιση του πίνακα κατά στήλες οι εντολές θα είναι:

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
    ΕΜΦΑΝΙΣΕ A[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

26. Να γραφεί αλγόριθμος ο οποίος θα δημιουργεί έναν τετραγωνικό πίνακα 5 x 5 που στις θέσεις με δείκτη γραμμής μικρότερο από τον δείκτη της στήλης θα καταχωρεί το μηδέν, στις θέσεις με δείκτη γραμμής ίσο με τον δείκτη στήλης θα καταχωρεί το 2 και στις υπόλοιπες θέσεις θα καταχωρεί τη μονάδα.

ΛΥΣΗ

Ένας πίνακας λέγεται τετραγωνικός όταν έχει ίδιο αριθμό γραμμών και στηλών. Τα στοιχεία που βρίσκονται στην κύρια διαγώνιο του πίνακα έχουν το χαρακτηριστικό ότι ο δείκτης γραμμής που βρίσκονται ισούται με το δείκτη της στήλης δηλαδή $i=j$. Τα στοιχεία της δευτερεύουσας διαγωνίου έχουν το χαρακτηριστικό ότι $i+j=N+1$, όπου $N=n$ διάσταση του πίνακα. Για τον παρακάτω τετραγωνικό πίνακα 5 X 5 για τα στοιχεία της δευτερεύουσας διαγωνίου ισχύει δηλαδή $i+j=6$. Επιπλέον, για τα στοιχεία πάνω από την κύρια διαγώνιο ισχύει $i<j$, ενώ για αυτά κάτω από την κύρια διαγώνιο $i>j$.

A[1,1]				A[1,5]	↗ Δευτερεύουσα διαγώνιος
	A[2,2]		A[2,4]		
		A[3,3]			
	A[4,2]		A[4,4]		
A[5,1]				A[5,5]	↘ Κύρια διαγώνιος

Για τη λύση του παραπάνω προβλήματος θα προσπελάσουμε όλα τα στοιχεία του τετραγωνικού πίνακα και θα καταχωρούμε την κατάλληλη τιμή, ανάλογα με το αποτέλεσμα της σύγκρισης των αντίστοιχων δεικτών. Ακολουθεί ο αλγόριθμος και ο πίνακας που θα προκύψει:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5
    ΑΝ i<j ΤΟΤΕ
      A[i,j]←0
    ΑΛΛΙΩΣ_ΑΝ i>j ΤΟΤΕ
      A[i,j]←1
    ΑΛΛΙΩΣ
      A[i,j]←2
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_Άσκηση

```

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

27. Να γραφεί αλγόριθμος που με δεδομένο έναν πίνακα A 4X4 και θα δημιουργεί έναν μονοδιάστατο πίνακα B 16 στοιχείων, που στις πρώτες 4 θέσεις του θα έχει τα στοιχεία της πρώτης γραμμής του πίνακα A, στις επόμενες 4 τα στοιχεία της δεύτερης γραμμής, κ.ο.κ.

ΛΥΣΗ

Για τη δημιουργία του πίνακα B θα πρέπει να προσπελάσουμε ένα-ένα τα στοιχεία του πίνακα A κατά γραμμές. Επίσης, θα χρησιμοποιήσουμε μία μεταβλητή που θα δείχνει σε ποια θέση του πίνακα B θα καταχωρίσουμε το αντίστοιχο στοιχείο του πίνακα A. Η μεταβλητή αυτή θα έχει αρχική τιμή=1 και σε κάθε προσπέλαση στοιχείου του A θα αυξάνεται κατά ένα για την επόμενη θέση του πίνακα B.

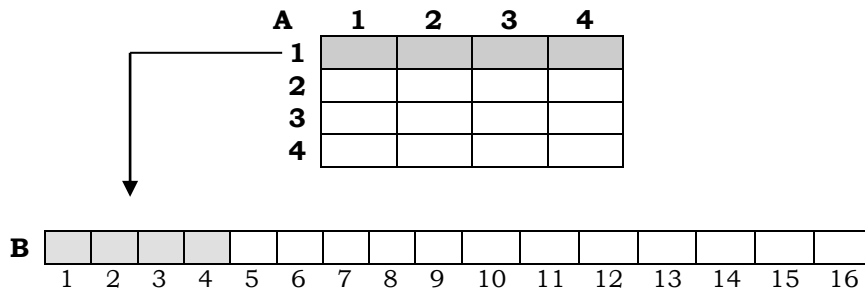
ΑΛΓΟΡΙΘΜΟΣ Άσκηση**ΔΕΔΟΜΕΝΑ** //A//

θέση ← 1

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 4**ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** 4

B[θέση] ← A[i,j]

θέση ← θέση + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ****ΤΕΛΟΣ** Άσκηση**ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ**

- 28. α.** Να μεταφέρετε στο τετράδιό σας τον πίνακα A και να συμπληρώσετε τις τιμές των στοιχείων του, όπως θα είναι μετά την εκτέλεση του παρακάτω τμήματος αλγορίθμου.

ΓΙΑ γ **ΑΠΟ** 1 **ΜΕΧΡΙ** 4**ΓΙΑ** δ **ΑΠΟ** 1 **ΜΕΧΡΙ** 4

A[γ,δ] ← 17 - (γ-1)*4 - δ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**

A

1				
2				
3				
4				
	1	2	3	4

- β.** Να γράψετε στο τετράδιό σας τον αριθμό για καθένα από τα κενά **1-2** και δίπλα την κατάλληλη έκφραση, ώστε το παρακάτω τμήμα αλγορίθμου να έχει ακριβώς το ίδιο αποτέλεσμα με το προηγούμενο, δηλαδή να εκχωρεί στα στοιχεία του πίνακα A τις ίδιες τιμές.

κ ← (1)

ΓΙΑ γ **ΑΠΟ** 1 **ΜΕΧΡΙ** 4**ΓΙΑ** δ **ΑΠΟ** 1 **ΜΕΧΡΙ** 4

A[γ,δ] ← κ

κ ← (2)

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**

- 29.** Για δεδομένο πίνακα N γραμμών και M στηλών να γράψετε αλγόριθμο ο οποίος θα εμφανίζει το πλήθος των στοιχείων του πίνακα για τα οποία τα δύο γειτονικά τους στην ίδια γραμμή στοιχεία είναι μικρότερα από αυτό.

- 30.** Να γραφεί αλγόριθμος που θα διαβάζει τα στοιχεία ενός μονοδιάστατου πίνακα 100 θέσεων και θα δημιουργεί έναν νέο δισδιάστατο πίνακα 10x10, όπου τα 10 πρώτα στοιχεία του μονοδιάστατου θα καταχωριστούν στην πρώτη γραμμή του διασδιάστατου, τα 10 επόμενα στη δεύτερη γραμμή, κ.ο.κ.

- 31.** Να γίνει αλγόριθμος ο οποίος θα διαβάζει έναν πίνακα 20x30 και έναν ακέραιο από 1 έως 20 με έλεγχο εγκυρότητας. Στη συνέχεια θα δημιουργεί έναν νέο πίνακα 19x30 ο οποίος θα περιέχει τα στοιχεία του πρώτου πίνακα εκτός της γραμμής που καθορίζεται από τον αριθμό που έδωσε ο χρήστης.

ΑΠΛΕΣ ΕΦΑΡΜΟΓΕΣ ΜΕ ΔΙΣΔΙΑΣΤΑΤΟΥΣ ΠΙΝΑΚΕΣ**ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ**

- 32.** Να γραφεί πρόγραμμα που θα διαβάζει τα ονόματα των 10 παικτών μιας ομάδας μπάσκετ καθώς και τους πόντους καθενός σε κάθε έναν από τους 30 αγώνες πρωταθλήματος. Στη συνέχεια να εμφανίζει για κάθε παίκτη το όνομά του και τους αγώνες που δεν πέτυχε κανέναν πόντο.

ΛΥΣΗ

Θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα χαρακτήρων 10 θέσεων για να αποθηκεύσουμε τα ονόματα των 10 παικτών και έναν δισδιάστατο πίνακα ακεραίων 10 x 30 για να αποθηκεύσουμε τους πόντους των παικτών:

	ΟΝ		Π	1	2		30
1		→	1				
2		→	2				
...		→	...				
10		→	10				

Οι πίνακες είναι παράλληλοι γιατί για παράδειγμα ο 2^{ος} παίκτης που το όνομά του βρίσκεται στη θέση 2 του πίνακα ΟΝ, έχει πετύχει τους πόντους που είναι καταχωρημένοι στη 2^η γραμμή του πίνακα Π. Δηλαδή οι γραμμές του πίνακα Π αντιστοιχούν στους παίκτες και οι στήλες στους αγώνες. Θα μπορούσαμε να χρησιμοποιήσουμε για την καταχώρηση των πόντων των παικτών και έναν πίνακα 30 x 10 όπου οι γραμμές θα αντιστοικούσαν στους αγώνες και οι στήλες στους παίκτες. Σε αυτή την περίπτωση θα σχεδιάζαμε τους πίνακές μας ως εξής:

	1	2		10
ΟΝ				
	↓	↓	↓	↓
Π	1	2		10
1				
2				
...				
30				

Προσέχουμε δηλαδή στο σχεδιασμό μας να φαίνεται η παραλληλία των πινάκων μας για διευκόλυνση στη επίλυση της άσκησης. Δεν θα ήταν για παράδειγμα σωστός ο παρακάτω σχεδιασμός:

	1	2		10
ΟΝ				
Π	1	2		30
1				
2				
...				
10				

Για το συγκεκριμένο πρόβλημα εμείς επιλέγουμε τον πρώτο σχεδιασμό. Αφού διαβάσουμε τους πίνακες, για να εμφανίσουμε για κάθε παίκτη τους αγώνες που δεν πέτυχε κανέναν πόντο, πρέπει να προσπελάσουμε το πίνακα Π κατά γραμμές, και αν βρίσκουμε μηδενικό στοιχείο να εμφανίζουμε τον αριθμό της στήλης που βρισκόμαστε (οι στήλες αντιστοιχούν στους αγώνες σύμφωνα με τον πρώτο σχεδιασμό).

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π[10,30],i,j

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[10]

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

ΔΙΑΒΑΣΕ ΟΝ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

ΔΙΑΒΑΣΕ Π[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

διάβασμα πρώτα του πίνακα ΟΝ και μετά του πίνακα Π. Μπορεί το διάβασμα των δύο πινάκων να γίνει ταυτόχρονα, δηλαδή για κάθε όνομα του παίκτη να διαβάζουμε και τους πόντους που πέτυχε στους 30 αγώνες

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

ΔΙΑΒΑΣΕ ΟΝ[i]

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

ΔΙΑΒΑΣΕ Π[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

ΓΡΑΨΕ ΟΝ[i]

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

ΑΝ Π[i,j]=0 **ΤΟΤΕ** **ΓΡΑΨΕ** j

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

33. Να υλοποιηθεί αλγόριθμος που θα διαβάζει το όνομα 50 προϊόντων που εξάγει η Ελλάδα σε 20 ευρωπαϊκές χώρες. Στη συνέχεια θα διαβάζει το όνομα των 20 αυτών χωρών και την ποσότητα σε τόνους κάθε προϊόντος που εξάγεται σε κάθε χώρα. Να εμφανίζει για κάθε χώρα το όνομά της και το όνομα των προϊόντων που έχει εξαχθεί πάνω από 2.000 τόνους. Επίσης να εμφανίζει για κάθε προϊόν το όνομά του και τις χώρες στις οποίες εξάχθηκε λιγότερο από 1.000 τόνους.

34. Στο ελληνικό πρωτάθλημα μπάσκετ συμμετέχουν 14 ομάδες, καθεμία από τις οποίες αποτελείται από 12 παίκτες. Να φτιάξετε πρόγραμμα το οποίο θα διαβάζει το ύψος των παικτών κάθε ομάδας και θα το αποθηκεύει σε πίνακα 14x12. Στη συνέχεια θα εμφανίζει το πλήθος των παικτών που έχουν ύψος πάνω από τα 5/6 του μέσου όρου ύψους όλων των παικτών του πρωταθλήματος.

ΑΘΡΟΙΣΜΑ ΓΡΑΜΜΩΝ ΚΑΙ ΣΤΗΛΩΝ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

35. Μια ομάδα πόλο αποτελείται από 12 παίκτες και δίνει 30 αγώνες. Να γίνει πρόγραμμα που θα διαβάζει για κάθε πολίστα της ομάδας το όνομά του και τα γκολ που πέτυχε σε καθένα από τους 30 αγώνες και θα εμφανίζει:

i) το όνομα κάθε παίκτη και τον μέσο όρο των γκολ που πέτυχε ανά αγώνα

ii) για κάθε αγώνα τον αύξοντα αριθμό του και το σύνολο των γκολ που πέτυχε ολόκληρη η ομάδα στον αγώνα αυτό.

ΛΥΣΗ

Θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα ΟΝ, 12 θέσεων για την αποθήκευση των ονομάτων των 12 παικτών και έναν διασδιάστατο πίνακα ΓΚ 12x30 για την αποθήκευση των γκολ κάθε παίκτη σε κάθε αγώνα. Στη συνέχεια, για το i) ερώτημα θα φτιάξουμε έναν μονοδιάστατο πίνακα ΜΟ 12 θέσεων, που κάθε στοιχείο του θα είναι ο μέσος όρος των γκολ κάθε παίκτη:

	ΟΝ	ΓΚ	1	2		30		ΜΟ
1		1	-----	-----	-----	-----	-->	1
2		2	-----	-----	-----	-----	-->	2
...		...						...
12		12	-----	-----	-----	-----	-->	12

Για τον υπολογισμό του 1^{ου} στοιχείου του πίνακα ΜΟ θα πρέπει να υπολογίσουμε το άθροισμα της 1^{ης} γραμμής του πίνακα ΓΚ και να διαιρέσουμε με το 30. Το άθροισμα της πρώτης γραμμής του ΓΚ θα είναι: $A\Theta = \Gamma K[1,1] + \Gamma K[1,2] + \Gamma K[1,3] + \dots + \Gamma K[1,30]$, οπότε για τον υπολογισμό του μπορούμε να γράψουμε τις παρακάτω εντολές:

$A\Theta \leftarrow 0$

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

$A\Theta \leftarrow A\Theta + \Gamma K[1,j]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

και για το ΜΟ[1] θα είναι $ΜΟ[1] \leftarrow A\Theta / 30$.

Για τον υπολογισμό του 2^{ου} στοιχείου του ΜΟ, δηλαδή του ΜΟ[2] οι εντολές θα είναι:

$A\Theta \leftarrow 0$

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

$A\Theta \leftarrow A\Theta + \Gamma K[2,j]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

$ΜΟ[2] \leftarrow A\Theta / 30$

Άρα για το υπολογισμό και των 12 στοιχείων του ΜΟ οι εντολές θα είναι:

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12

$A\Theta \leftarrow 0$

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

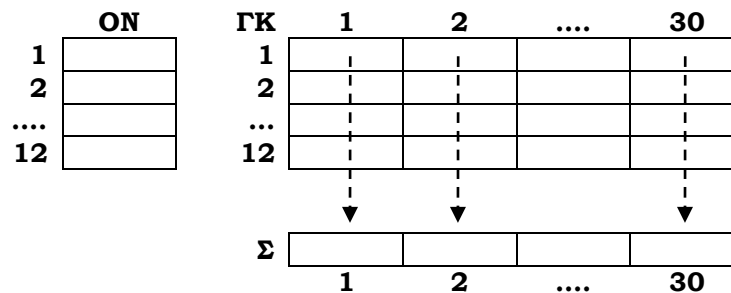
$A\Theta \leftarrow A\Theta + \Gamma K[i,j]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

$ΜΟ[i] \leftarrow A\Theta / 30$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Για το ερώτημα ii) θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα Σ 30 θέσεων παράλληλο με τις στήλες του πίνακα, που κάθε του στοιχείο θα είναι το άθροισμα των γκολ όλων των παικτών για έναν αγώνα. Δηλαδή το Σ[1] θα είναι το άθροισμα της πρώτης στήλης του πίνακα ΓΚ, το Σ[2] το άθροισμα της 2^{ης}, κ.ο.κ.



Το $\Sigma[1]$ θα είναι $\Sigma[1] = \Gamma K[1,1] + \Gamma K[2,1] + \Gamma K[3,1] + \dots + \Gamma K[12,1]$, οπότε για τον υπολογισμό του θα γράψουμε:

```

Σ[1] ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
    Σ[1] ← Σ[1] + ΓΚ[i,1]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Άρα για τον υπολογισμό και των τριάντα στοιχείων του Σ οι εντολές θα είναι:

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
    Σ[j] ← 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
        Σ[j] ← Σ[j] + ΓΚ[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Το συνολικό πρόγραμμα θα είναι:

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i,j,AΘ,ΓΚ[12,30],Σ[30]

ΧΑΡΑΚΤΗΡΕΣ: ON[12]

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ[12]

ΑΡΧΗ

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
    ΔΙΑΒΑΣΕ ON[i]
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
        ΔΙΑΒΑΣΕ ΓΚ[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

} διάβασμα των πινάκων

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΘ ← 0
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
        ΑΘ ← ΑΘ + ΓΚ[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ[i] ← ΑΘ / 30
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

} υπολογισμός των μέσων όρων

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
    ΓΡΑΨΕ ON[i], ΜΟ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

} εμφάνιση ονομάτων και μέσων όρων (ερώτημα i)

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
    Σ[j] ← 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
        Σ[j] ← Σ[j] + ΓΚ[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

} υπολογισμός αθροισμάτων

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
    ΓΡΑΨΕ j, Σ[j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

} εμφάνιση αγώνων και αθροισμάτων (ερώτημα ii)

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

36. Σε έναν διαγωνισμό γνώσεων 40 μαθητές ρωτήθηκαν για το έτος που διεξήχθησαν 30 γεγονότα από την Παγκόσμια Ιστορία. Να γράψετε αλγόριθμο ο οποίος να διαβάζει αρχικά για τα 30 παγκόσμια γεγονότα το έτος που διεξήχθησαν. Στη συνέχεια για κάθε μαθητή να διαβάζει το όνομα του καθώς και το έτος που έδωσε ως απάντηση για καθένα από τα 30 γεγονότα. Ο αλγόριθμος να εμφανίζει:

- i. για κάθε μαθητή το όνομα του και το πλήθος των χρονολογιών που έδωσε σωστά,
- ii. αν υπάρχει ιστορικό γεγονός που το απάντησαν σωστά όλοι οι μαθητές.

37. Σε ένα πρωτάθλημα ποδοσφαίρου συμμετέχουν 20 ομάδες, που καθεμία από αυτές δίνει 38 αγώνες. Να υλοποιήσετε πρόγραμμα το οποίο θα διαβάζει το όνομα των ομάδων και το γράμμα «N» ή το γράμμα «H» ή το γράμμα «I» (Νίκη, Ήττα, Ισοπαλία αντίστοιχα) για καθέναν από τους αγώνες που έδωσε, κάνοντας παράλληλα έλεγχο εγκυρότητας των τιμών αυτών. Αν η νίκη δίνει τρεις βαθμούς στην ομάδα, η ισοπαλία έναν και η ήττα μηδέν, το πρόγραμμα να εμφανίζει για κάθε ομάδα το όνομα της, τη συνολική της βαθμολογία και το πλήθος των νικών της.

38. Το χρημασθηριακό γραφείο Μαρματάκης στα Χανιά έχει 230 εγγεγραμμένους πελάτες. Κάθε πελάτης κατέχει έναν αριθμό μετοχών από τις 180 εταιρείες που διαπραγματεύονται στο Χρηματιστήριο Αθηνών. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει για κάθε εταιρεία το όνομα της και την αξία της μετοχής της. Στη συνέχεια ο αλγόριθμος θα διαβάζει το όνομα κάθε πελάτη του γραφείου καθώς και τον αριθμό μετοχών που κατέχει για καθεμία από τις εταιρείες του χρημασθηρίου. Παράλληλα να κάνει και έλεγχο εγκυρότητας των αριθμών των μετοχών, ώστε να είναι θετικός αριθμός ή μηδέν. Ο αλγόριθμος να εμφανίζει:

- i. για κάθε πελάτη το όνομα του και πόσα χρήματα έχει σε μετοχές και
- ii. για κάθε εταιρεία το όνομα της και το πλήθος των μετοχών που κατέχουν όλοι οι πελάτες του γραφείου Μαρματάκης.

ΜΕΓΙΣΤΟ ΚΑΙ ΕΛΑΧΙΣΤΟ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟΥΣ ΠΙΝΑΚΕΣ **ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ**

39. Μια εταιρεία αποτελείται από 10 υποκαταστήματα σε όλη την Ελλάδα. Να γίνει πρόγραμμα που θα διαβάζει για κάθε υποκατάστημα τα μηνιαία έσοδά του για ένα χρόνο καθώς και το όνομα του νομού στον οποίο βρίσκεται. Θεωρώντας ότι τα υποκαταστήματα βρίσκονται σε διαφορετικούς νομούς, το πρόγραμμα να εμφανίζει σε ποιον νομό βρίσκεται το υποκατάστημα που έγινε το ρεκόρ εσόδων καθώς και τον μήνα που έγινε το ρεκόρ αυτό.

ΛΥΣΗ

Θα χρησιμοποιήσουμε καταρχήν έναν μονοδιάστατο πίνακα ON_NOM 10 θέσεων για την αποθήκευση των ονομάτων των νομών που βρίσκονται τα υποκαταστήματα και έναν δισδιάστατο πίνακα ΕΣ 10x12 που θα αποθηκευτούν τα μηνιαία έσοδα για κάθε υποκατάστημα. Άρα οι γραμμές του δισδιάστατου πίνακα θα αντιστοιχούν στα υποκαταστήματα και οι στήλες στους μήνες. Η μορφή των πινάκων μας θα είναι η ακόλουθη:

	ON_NOM	ΕΣ	1	2		12
1		1				
2		2				
...		...				
10		10				

Το πρόγραμμα ζητάει να βρούμε όχι το μέγιστο στοιχείο του πίνακα ΕΣ, αλλά τη θέση του πίνακα ΕΣ που βρίσκεται το μεγαλύτερο στοιχείο. Αν για παράδειγμα βρούμε ότι το μέγιστο στοιχείο του πίνακα βρίσκεται στη θέση 3,8, θα ξέρουμε ότι στο 3^ο υποκατάστημα τον 8^ο μήνα σημειώθηκαν οι μέγιστες πωλήσεις. Οπότε θα πρέπει να εμφανίσουμε το όνομα του νομού, δηλαδή το ON[3] και τον αριθμό του μήνα δηλαδή το 8. Για την εύρεση του μέγιστου στοιχείου και της θέσης στην οποία βρίσκεται, υποθέτουμε ότι το μεγαλύτερο στοιχείο βρίσκεται στη θέση 1,1 του πίνακα, οπότε αρχικοποιούμε 3 μεταβλητές ως εξής:

max←ΕΣ[1,1]
γραμμή_max←1
στήλη_max←1

Στη συνέχεια συγκρίνουμε ένα-ένα όλα τα στοιχεία του πίνακα ΕΣ με το max, και αν βρούμε στοιχείο μεγαλύτερο, στη μεταβλητή max εκχωρούμε το στοιχείο αυτό (max←ΕΣ[i,j]), στη μεταβλητή γραμμή_max εκχωρούμε τη γραμμή στην οποία βρισκόμαστε (γραμμή_max←i) και στη μεταβλητή στήλη_max εκχωρούμε τη στήλη στην οποία βρισκόμαστε (στήλη_max←j). Το ολοκληρωμένο πρόγραμμα έχει ως εξής:

ΠΡΟΓΡΑΜΜΑ Άσκηση**ΜΕΤΑΒΛΗΤΕΣ****ΑΚΕΡΑΙΕΣ:** ΕΣ[10,12],i,j,max,γραμμή_max,στήλη_max**ΧΑΡΑΚΤΗΡΕΣ:** ON_NOM[10]**ΑΡΧΗ**

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
    ΔΙΑΒΑΣΕ ON_NOM[i]
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 12
        ΔΙΑΒΑΣΕ ΕΣ[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

διάβασμα των πινάκων

```

max←ΕΣ[1,1]
γραμμή_max←1
στήλη_max←1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 12
        ΑΝ ΕΣ[i,j]>max ΤΟΤΕ
            max←ΕΣ[i,j]
            γραμμή_max←i
            στήλη_max←j
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

υπολογισμός μέγιστων εσόδων και θέσης τους στον πίνακα

ΓΡΑΨΕ ON_NOM[γραμμή_max],στήλη_max !εμφάνιση των ζητούμενων**ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ**

40. Στον τελικό του άλματος εις μήκος συμμετέχουν 12 αθλητές και κάθε αθλητής κάνει 6 άλματα. Ως τελική επίδοση του αθλητή λαμβάνεται το μεγαλύτερο άλμα από τα έξι. Να γραφεί αλγόριθμος που θα διαβάζει για κάθε αθλητή το όνομά του και την επίδοσή του σε καθένα από τα 6 άλματα. Αν κάποια προσπάθειά του είναι άκυρη, τότε καταχωρείται για την συγκεκριμένη προσπάθεια ένας αρνητικός αριθμός. Ο αλγόριθμος να εμφανίζει για κάθε αθλητή, το όνομά του και την τελική του επίδοση. Αν κάποιος αθλητής δεν έχει κάνει κανένα έγκυρο άλμα, τότε γι'αυτόν να εμφανίζει το όνομά του και το μήνυμα «καμία έγκυρη προσπάθεια».

ΛΥΣΗ

Θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα ON 12 θέσεων για τα ονόματα των 12 αθλητών και έναν διοδιάστατο πίνακα ΕΠΙΑ 12x6 για τις επιδόσεις των αθλητών. Επίσης, θα φτιάξουμε και έναν μονοδιάστατο πίνακα ΜΕΓ 12 θέσεων για τον υπολογισμό της καλύτερης προσπάθειας κάθε αθλητή. Οι πίνακες θα έχουν την παρακάτω μορφή:

	ON	ΕΠΙΑ	1	2	...	6	ΜΕΓ
1		1					1
2		2					2
...		...					...
12		12					12

Για τον υπολογισμό της καλύτερης προσπάθειας του 1^{ου} αθλητή, δηλαδή του στοιχείου ΜΕΓ[1], οι εντολές θα είναι:

```

ΜΕΓ[1]←ΕΠΙΑ[1,1]
ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 6
    ΑΝ ΕΠΙΑ[1,j]>ΜΕΓ[1] ΤΟΤΕ
        ΜΕΓ[1]←ΕΠΙΑ[1,j]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

σκεφτόμαστε σαν να πρόκειται για μονοδιάστατο πίνακα, δηλαδή υποθέτουμε ότι μεγαλύτερο είναι το 1^ο στοιχείο του πίνακα δηλ. το ΕΠΙΑ[1,1]. Συγκρίνουμε στη συνέχεια ένα ένα όλα τα στοιχεία της πρώτης γραμμής ξεκινώντας από το 2^ο δηλ. το ΕΠΙΑ[1,2] και αν βρούμε στοιχείο μεγαλύτερο, εκχωρούμε στο ΜΕΓ[1] το στοιχείο αυτό.

Για τον υπολογισμό της καλύτερης προσπάθειας του 2^{ου} αθλητή, δηλαδή του στοιχείου ΜΕΓ[2], οι εντολές θα είναι:

```

ΜΕΓ[2]←ΕΠΙΔ[2,1]
ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 6
  ΑΝ ΕΠΙΔ[2, j]> ΜΕΓ[2] ΤΟΤΕ
    ΜΕΓ[2]←ΕΠΙΔ[2, j]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Οπότε, οι εντολές υπολογισμού της καλύτερης επίδοσης για κάθε αθλητή από τον 1^ο ως τον 12^ο θα είναι:

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
  ΜΕΓ[i]←ΕΠΙΔ[i,1]
  ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 6
    ΑΝ ΕΠΙΔ[i, j]> ΜΕΓ[i] ΤΟΤΕ
      ΜΕΓ[i]←ΕΠΙΔ[i, j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Στη συνέχεια, πρέπει να εμφανίσουμε για κάθε αθλητή το όνομά του και την καλύτερή του προσπάθεια, δηλαδή τα στοιχεία του πίνακα ΟΝ και ΜΕΓ. Θα πρέπει όμως αν βρίσκουμε κάποιο στοιχείο του πίνακα ΜΕΓ αρνητικό να εμφανίζουμε το μήνυμα «καμία έγκυρη προσπάθεια». Ο ολοκληρωμένος αλγόριθμος θα είναι:

```

ΑΛΓΟΡΙΘΜΟΣ Άσκηση
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
  ΔΙΑΒΑΣΕ ΟΝ[i]
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 6
    ΔΙΑΒΑΣΕ ΕΠΙΔ[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
    ΜΕΓ[i]←ΕΠΙΔ[i,1]
    ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 6
      ΑΝ ΕΠΙΔ[i, j]> ΜΕΓ[i] ΤΟΤΕ
        ΜΕΓ[i]←ΕΠΙΔ[i, j]
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

διάβασμα των πινάκων

υπολογισμός της καλύτερης επίδοσης κάθε αθλητή

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
  ΕΜΦΑΝΙΣΕ ΟΝ[i]
  ΑΝ ΜΕΓ[i]>=0 ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ ΜΕΓ[i]
  ΑΛΛΙΩΣ
    ΕΜΦΑΝΙΣΕ 'Καμία έγκυρη προσπάθεια'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ Άσκηση

```

για κάθε αθλητή εμφάνισε το όνομά του και αν η καλύτερή του επίδοση είναι θετική εμφάνισέ τη, αλλιώς εμφάνισε το μήνυμα

41. Να γραφεί αλγόριθμος που θα διαβάζει το όνομα 10 χωρών και για καθεμία απ'αυτές θα διαβάζει το όνομα των 15 μεγαλύτερων πόλεων της. Στη συνέχεια, για κάθε πόλη να διαβάζει τον πληθυσμό της και να εμφανίζει:

- τον μεγαλύτερο πληθυσμό
- το όνομα ή τα ονόματα των πόλεων με τον μεγαλύτερο πληθυσμό
- τις χώρες που έχουν έστω και μία πόλη με τον μεγαλύτερο πληθυσμό
- αν θεωρήσουμε ότι κάθε χώρα αποτελείται μόνο απ'αυτές τις 15 πόλεις, να εμφανίζει τη χώρα με τον μεγαλύτερο πληθυσμό.

ΛΥΣΗ

Θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα ΧΩΡ 10 θέσεων για την καταχώριση των ονομάτων των 10 χωρών, έναν δισδιάστατο πίνακα ΠΟΛ 15x10 για την αποθήκευση των ονομάτων της κάθε πόλης και έναν δισδιάστατο πίνακα ΠΛΗΘ 15x10 για την αποθήκευση των αντίστοιχων πληθυσμών:

	1	2		10
ΧΩΡ				

ΠΟΛ	1	2	...	10
1				
2				
...				
15				

ΠΛΗΘ	1	2	...	10
1				
2				
...				
15				

Αφού διαβάσουμε τους πίνακες, μπορούμε εύκολα να βρούμε τον μεγαλύτερο πληθυσμό, δηλαδή το μεγαλύτερο στοιχείο του πίνακα ΠΛΗΘ(ερώτημα i). Για το ερώτημα ii) θα πρέπει να προσπελάσουμε ένα-ένα όλα τα στοιχεία του πίνακα ΠΛΗΘ και όταν βρίσκουμε στοιχείο ίσο με το μεγαλύτερο πληθυσμό που υπολογίσαμε στο 1^ο ερώτημα, να εμφανίζουμε το αντίστοιχο στοιχείο του πίνακα ΠΟΛ.

Για την εμφάνιση των χωρών που έχουν τον μεγαλύτερο πληθυσμό, θα προσπελάσουμε τον πίνακα ΠΛΗΘ κατά στήλες και αν βρούμε στοιχείο ίδιο με το μεγαλύτερο πληθυσμό, θα πρέπει να εμφανίσουμε το όνομα της αντίστοιχης χώρας. Δηλαδή, για την πρώτη στήλη του πίνακα ΠΛΗΘ θα γράφαμε:

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
  ΑΝ ΠΛΗΘ[i,1]=max ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ ΧΩΡ[1]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Το πρόβλημα με τις παραπάνω εντολές είναι ότι αν μία χώρα έχει περισσότερες από μία πόλεις με τον μεγαλύτερο πληθυσμό, το όνομά της θα εμφανιστεί παραπάνω της μίας φορές. Για παράδειγμα, αν η πρώτη χώρα έχει την 2^η, την 8^η και την 10^η πόλη της με πληθυσμό ίσο με τον μέγιστο, το όνομά της θα εμφανιστεί 3 φορές, πράγμα που δεν είναι επιθυμητό. Γι'αυτό το λόγο τροποποιούμε τις παραπάνω εντολές ως εξής:

```

υπάρχει←ψευδής
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
  ΑΝ ΠΛΗΘ[i,1]=max ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ ΧΩΡ[1]
    υπάρχει←αληθής
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ υπάρχει=αληθής ΤΟΤΕ
  ΕΜΦΑΝΙΣΕ ΧΩΡ[1]
ΤΕΛΟΣ_ΑΝ

```

Οπότε τροποποιώντας τις παραπάνω εντολές για όλες τις χώρες, δηλ. για όλες τις στήλες του πίνακα ΠΛΗΘ, έχουμε:

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
  υπάρχει←ψευδής
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
    ΑΝ ΠΛΗΘ[i,j]=max ΤΟΤΕ
      υπάρχει←αληθής
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ υπάρχει=αληθής ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ ΧΩΡ[j]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Για το τελευταίο ερώτημα πρέπει να βρούμε το άθροισμα κατά στήλες του πίνακα ΠΛΗΘ και θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα Σ 10 θέσεων. Κατόπιν, θα βρούμε το μεγαλύτερο στοιχείο του πίνακα αυτού και την θέση του μέσα στον πίνακα και θα εμφανίσουμε το αντίστοιχο στοιχείο του πίνακα ΧΩΡ. Ο ολοκληρωμένος αλγόριθμος θα είναι:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
  ΔΙΑΒΑΣΕ ΧΩΡ[j]
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
    ΔΙΑΒΑΣΕ ΠΟΛ[i, j]
    ΔΙΑΒΑΣΕ ΠΛΗΘ[i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

} διάβασμα των πινάκων

```
max ← ΠΛΗΘ[1, 1]
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    ΑΝ ΠΛΗΘ[i, j] > max ΤΟΤΕ
      max ← ΠΛΗΘ[i, j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΕΜΦΑΝΙΣΕ max
```

} υπολογισμός και εμφάνιση του μεγαλύτερου πληθυσμού (ερώτημα i)

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    ΑΝ ΠΛΗΘ[i, j] = max ΤΟΤΕ
      ΕΜΦΑΝΙΣΕ ΠΟΛ[i, j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

} εμφάνιση των πόλεων με τον μεγαλύτερο πληθυσμό (ερώτημα ii)

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
  υπάρχει ← ψευδής
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
    ΑΝ ΠΛΗΘ[i, j] = max ΤΟΤΕ
      υπάρχει ← αληθής
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ υπάρχει = αληθής ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ ΧΩΡ[j]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

} ερώτημα iii

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
  Σ[j] ← 0
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
    Σ[j] ← Σ[j] + ΠΛΗΘ[i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

} υπολογισμός του συνόλου του πληθυσμού κάθε χώρας στον πίνακα Σ

```
max1 ← Σ[1]
θέση ← 1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10
  ΑΝ Σ[i] > max1 ΤΟΤΕ
    max1 ← Σ[i]
    θέση ← i
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

} υπολογισμός του μεγαλύτερου στοιχείου στον πίνακα Σ και της θέσης του στον πίνακα

```
ΕΜΦΑΝΙΣΕ ΧΩΡ[θέση]
ΤΕΛΟΣ_Άσκηση
```

! εμφάνιση της χώρας με τον μεγαλύτερο πληθυσμό

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

42. Μια στατιστική εξέταση το ύψος των μαθητών 30 σχολείων της επαρχίας. Αν κάθε σχολείο αποτελείται από 150 μαθητές, να γράψετε αλγόριθμο που θα διαβάζει για κάθε σχολείο το όνομα, το φύλο ("Α" για

αγόρι και "Κ" για κορίτσι) και το ύψος των 150 μαθητών τους, ελέγχοντας παράλληλα την εγκυρότητα του φύλου και του ύψους, ώστε αυτό να είναι θετικός αριθμός. Στη συνέχεια ο αλγόριθμος θα εμφανίζει:

- i. το όνομα του πιο ψηλού κοριτσιού και το όνομα του πιο ψηλού αγοριού από όλα τα σχολεία,
- ii. για κάθε σχολείο το όνομα του πιο ψηλού μαθητή του.

Παρατήρηση: Θεωρήστε ότι σε κάθε σχολείο υπάρχουν τόσο αγόρια όσο και κορίτσια.

43. Ένας διαγωνισμός τραγουδιού διεξάγεται σε δύο φάσεις.

Στην πρώτη φάση γίνεται ακρόαση των 45 τραγουδιών που διαγωνίζονται και κάθε μέλος της επταμελούς κριτικής επιτροπής βαθμολογεί το κάθε τραγούδι με βαθμό από 1 έως 10.

Στη δεύτερη φάση προκρίνεται κάθε τραγούδι που συγκέντρωσε συνολική βαθμολογία μεγαλύτερη του 50 και το οποίο όλοι οι κριτές έχουν βαθμολογήσει τουλάχιστον με 5.

Να γραφεί αλγόριθμος, ο οποίος:

Δ1. Για κάθε τραγούδι να διαβάζει τον τίτλο του και τον βαθμό που έδωσε κάθε κριτής. Δεν απαιτείται έλεγχος εγκυρότητας.

Δ2. Να υπολογίζει και να εμφανίζει τη συνολική βαθμολογία του κάθε τραγουδιού, η οποία προκύπτει ως το άθροισμα των βαθμών όλων των κριτών.

Δ3. Να βρίσκει και να εμφανίζει τους τίτλους των τραγουδιών που προκρίνονται στη δεύτερη φάση του διαγωνισμού. Αν κανένα τραγούδι δεν προκρίνεται στη δεύτερη φάση, να εμφανίζει κατάλληλο μήνυμα.

Δ4. Να βρίσκει και να εμφανίζει το πλήθος των κριτών που έδωσαν τον μέγιστο βαθμό τους σε ένα μόνο τραγούδι.

(εξετάσεις 2015)

44. Ένας ψαράς καταγράφει κάθε μήνα και για έναν χρόνο τον αριθμό των ψαριών που πιάνει για καθένα από τα 20 πιο συνηθισμένα ψάρια της χώρας. Να υλοποιήσετε αλγόριθμο ο οποίος θα διαβάζει για καθένα από τα 20 είδη ψαριών το όνομα του. Στη συνέχεια θα διαβάζει για κάθε μήνα του χρόνου για κάθε είδος τον αριθμό ψαριών που κατέγραψε ο ψαράς. Ο αλγόριθμος θα εμφανίζει:

1. το πιο συνηθισμένο είδος του ψαριού που ο ψαράς είχε στα δίχτυα του τη χρονιά που πέρασε.
2. το όνομα του είδους του ψαριού και τον αύξοντα αριθμό του μήνα για τον οποίο ο ψαράς είχε τη χειρότερη ψαριά.
3. για κάθε είδος ψαριού το όνομα του και τον αύξοντα αριθμό του μήνα που ο ψαράς είχε την καλύτερη ψαριά γι' αυτό.

ΠΡΟΒΛΗΜΑΤΑ ΜΕ ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

45. Μια ασφαλιστική εταιρεία έχει καταγράψει για καθένα από τους 10 ασφαλιστές της το όνομά του και τον αριθμό συμβολαίων που έκανε κάθε μήνα του έτους 2008. Να γίνει πρόγραμμα που αρχικά θα διαβάζει το όνομα κάθε ασφαλιστή και τον αριθμό συμβολαίων του για κάθε μήνα. Στη συνέχεια να διαβάζει ένα όνομα ασφαλιστή και αν αυτό υπάρχει στην εταιρεία, να εμφανίζει το πλήθος των συμβολαίων του για το έτος 2008, ενώ αν δεν υπάρχει να εμφανίζει διευκρινιστικό μήνυμα. Θεωρείστε ότι όλοι οι ασφαλιστές έχουν διαφορετικό όνομα.

ΛΥΣΗ

Θα χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα 10 θέσεων για την καταχώριση των ονομάτων των 10 ασφαλιστών και έναν παράλληλο δισδιάστατο πίνακα 10x12 (10 ασφαλιστές x 12 μήνες), για την αποθήκευση των αριθμών συμβολαίων των ασφαλιστών για κάθε μήνα του έτους:

	ΟΝ	ΣΥΜΒ	1	2		12
1		1				
2		2				
....						
10		10				

Στη συνέχεια, αφού διαβάσουμε το όνομα του ασφαλιστή που θα δώσει ο χρήστης, θα εφαρμόσουμε τον αλγόριθμο της σειριακής αναζήτησης γι' αυτό το όνομα, στον πίνακα με τα ονόματα των ασφαλιστών. Αν βρεθεί το συγκεκριμένο όνομα, θα προσπελάσουμε τα στοιχεία της αντίστοιχης γραμμής του πίνακα ΣΥΜΒ για να βρούμε το πλήθος των συμβολαίων που έκανε ο ασφαλιστής:

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j, ΘΕΣΗ, ΠΛ_ΣΥΜΒ, ΣΥΜΒ[10,12]

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[10], ΟΝ_ΑΣΦ

ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΔΙΑΒΑΣΕ ΟΝ[i]

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 12

ΔΙΑΒΑΣΕ ΣΥΜΒ[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} διάβασμα των πινάκων

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ON_ΑΣΦ ! διάβασμα του ονόματος ασφαλιστή προς αναζήτηση

BPEΘΗΚΕ←ΨΕΥΔΗΣ

ΘΕΣΗ←0

i←1

ΟΣΟ i<=10 **ΚΑΙ** BPEΘΗΚΕ=ΨΕΥΔΗΣ **ΕΠΑΝΑΛΑΒΕ**

ΑΝ ON[i]=ON_ΑΣΦ **ΤΟΤΕ**

BPEΘΗΚΕ←ΑΛΗΘΗΣ

ΘΕΣΗ←i

ΑΛΛΙΩΣ

i←i+1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ BPEΘΗΚΕ=ΨΕΥΔΗΣ **ΤΟΤΕ**

ΓΡΑΨΕ 'Δεν βρέθηκε ο ασφαλιστής που δώσατε'

ΑΛΛΙΩΣ

ΠΛ_ΣΥΜΒ←0

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 12

ΠΛ_ΣΥΜΒ←ΠΛ_ΣΥΜΒ+ΣΥΜΒ[ΘΕΣΗ, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ΠΛ_ΣΥΜΒ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

αναζήτηση του ονόματος στον πίνακα ON

αν έχει βρεθεί ο ασφαλιστής στη μεταβλητή ΘΕΣΗ, προσθέτουμε όλα τα στοιχεία της αντίστοιχης γραμμής του πίνακα ΣΥΜΒ στη μεταβλητή ΠΛ_ΣΥΜΒ

46. Τον τελευταίο χρόνο διοργανώθηκαν 40 διαγωνισμοί σκακιού σε 40 διαφορετικές χώρες του κόσμου. Σε κάθε διαγωνισμό συμμετείχαν 30 παίκτες, όχι αναγκαστικά οι ίδιοι σε κάθε διαγωνισμό. Να γίνει αλγόριθμος που θα διαβάζει τα ονόματα των χωρών που έγιναν οι διαγωνισμοί και για κάθε χώρα τα ονόματα των παικτών που συμμετείχαν στο διαγωνισμό της καθώς και την βαθμολογία που συγκέντρωσε ο καθένας. Στη συνέχεια, να διαβάζει το όνομα ενός παίκτη και να εμφανίζει τις χώρες όπου ο παίκτης συμμετείχε στο διαγωνισμό τους και τη συνολική βαθμολογία που συγκέντρωσε από όλους τους διαγωνισμούς. Αν το όνομα αυτό δεν συμμετείχε σε κανέναν διαγωνισμό, να εμφανίζει κατάλληλο μήνυμα.

ΛΥΣΗ

Θα χρειαστούμε έναν μονοδιάστατο πίνακα ON_X 40 θέσεων για την αποθήκευση των ονομάτων των χωρών, και δύο παράλληλους δισδιάστατους πίνακες 40x30, τον έναν για την αποθήκευση των ονομάτων των παικτών (40 χώρες X 30 παίκτες πίνακας ON_Π), και έναν 40x30 για την αντίστοιχη βαθμολογία του κάθε παίκτη (πίνακας ΒΑΘ):

ON_X		ON_Π	1	2	...	30	ΒΑΘ	1	2	...	30
1		1					1				
2		2					2				
...		...					...				
40		40					40				

Αφού διαβάσουμε τους πίνακες, θα διαβάσουμε και το όνομα του προς αναζήτηση παίκτη. Επειδή τα ονόματα δεν είναι μοναδικά, θα προσπελάσουμε όλες τις θέσεις του πίνακα ON_Π και δεν θα σταματήσουμε μόλις βρούμε το όνομα του παίκτη. Κάθε φορά που θα βρίσκουμε το όνομά του, θα εμφανίζουμε το αντίστοιχο στοιχείο του πίνακα ON_X με τα ονόματα των χωρών, και θα προσθέτουμε σε έναν αθροιστή το αντίστοιχο στοιχείο του πίνακα ΒΑΘ με τις βαθμολογίες:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 40

ΔΙΑΒΑΣΕ ON_X[i]

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΔΙΑΒΑΣΕ ON_Π[i, j], ΒΑΘ[i, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

διάβασμα των πινάκων

ΔΙΑΒΑΣΕ ΟΝΟΜΑ ! διάβασμα του ονόματος του παίκτη που αναζητάμε

ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ

ΣΥΝ_ΒΑΘ←0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 40

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30

ΑΝ ON_Π[i, j]=ΟΝΟΜΑ **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ ON_Χ[i]

ΣΥΝ_ΒΑΘ←ΣΥΝ_ΒΑΘ+ΒΑΘ[i, j]

ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

κάνουμε προσπέλαση σε όλα τα στοιχεία του πίνακα ON_Π και αν κάποιο στοιχείο ισούται με το όνομα που ψάχνουμε, τότε εμφανίζουμε το ΑΝΤΙΣΤΟΙΧΟ στοιχείο του πίνακα ON_ΧΩΡ, προσθέτουμε στον αθροιστή τη βαθμολογία του παίκτη και αλλάζουμε το ΒΡΕΘΗΚΕ σε ΑΛΗΘΗΣ, έτσι ώστε να ξέρουμε όταν βρούμε από το βρόγχο επανάληψης αν το όνομα έχει βρεθεί ή όχι

ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ 'Δεν βρέθηκε ο παίκτης'

ΑΛΛΙΩΣ

ΕΜΦΑΝΙΣΕ ΣΥΝ_ΒΑΘ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ Άσκηση

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

47. Μια εταιρεία Πληροφορικής καταγράφει, για δέκα ιστότοπους, τον αριθμό των επισκέψεων που δέχεται ο καθένας, κάθε μέρα, για τέσσερις εβδομάδες.

Να αναπτύξετε αλγόριθμο, ο οποίος:

Δ1. Για καθένα από τους ιστότοπους να διαβάξει το όνομά του και τον αριθμό των επισκέψεων που δέχθηκε ο ιστότοπος για καθεμία ημέρα. Δεν απαιτείται έλεγχος εγκυρότητας τιμών.

Δ2. Να εμφανίζει το όνομα κάθε ιστότοπου και τον συνολικό αριθμό των επισκέψεων που δέχθηκε αυτός στο διάστημα των τεσσάρων εβδομάδων.

Δ3. Να εμφανίζει τα ονόματα των ιστοτόπων που κάθε μέρα στο διάστημα των τεσσάρων εβδομάδων δέχθηκαν περισσότερες από 500 επισκέψεις. Αν δεν υπάρχουν τέτοιοι ιστότοποι, να εμφανίζει κατάλληλο μήνυμα.

Δ4. Να διαβάξει το όνομα ενός ιστοτόπου. Αν το όνομα αυτό δεν είναι ένα από τα δέκα ονόματα που έχουν δοθεί, να το ξαναζητά, μέχρι να δοθεί ένα από αυτά τα ονόματα. Να εμφανίζει τους αριθμούς των εβδομάδων (1-4) κατά τη διάρκεια των οποίων ο συνολικός (εβδομαδιαίος) αριθμός επισκέψεων στον ιστότοπο αυτό είχε τη μέγιστη τιμή. (εξετάσεις 2014)

48. Για την παρακολούθηση των θερμοκρασιών της επικράτειας κατά τον μήνα Μάιο καταγράφεται κάθε ημέρα η θερμοκρασία στις 12:00 το μεσημέρι για 20 πόλεις. Να σχεδιάσετε αλγόριθμο που:

α. θα διαβάξει τα ονόματα των 20 πόλεων και τις αντίστοιχες θερμοκρασίες για καθεμία από τις ημέρες του μήνα και θα καταχωρίζει τα στοιχεία σε πίνακες,

β. θα διαβάξει το όνομα μιας πόλης και θα εμφανίζει τη μέγιστη θερμοκρασία της στη διάρκεια του μήνα. Αν δεν υπάρχει η πόλη στον πίνακα, θα εμφανίζει κατάλληλα διαμορφωμένο μήνυμα,

γ. θα εμφανίζει το πλήθος των ημερών που η μέση θερμοκρασία των 20 πόλεων ξεπέρασε τους 20 °C, αλλά όχι τους 30 °C. (Εξετάσεις 2006)

49. Μια τράπεζα έχει καταγράψει για κάθε πελάτη της το όνομα του, τον αριθμό ταυτότητας του και το ποσό που έχει στον λογαριασμό του. Να γράψετε αλγόριθμο ο οποίος, αφού διαβάσει τις πληροφορίες για 100 πελάτες, θα εμφανίζει το παρακάτω μενού επιλογής, υποχρεώνοντας τον χρήστη να δώσει μία τιμή από 1 έως 3.

1. αναζήτηση με βάση το όνομα

2. αναζήτηση με βάση τον αριθμό ταυτότητας

3. έξοδος

Δώσε μία τιμή από 1 έως 3.

Αν για παράδειγμα ο χρήστης δώσει την τιμή 1, τότε ο αλγόριθμος θα πρέπει να ζητάει από τον χρήστη το όνομα προς αναζήτηση και θα του εμφανίζει το ποσό που έχει στον λογαριασμό του. Αν το όνομα δεν υπάρχει, θα εμφανίζει κατάλληλο μήνυμα. Στη συνέχεια θα εμφανίζει πάλι το μενού και θα ζητάει πάλι από τον χρήστη να διαλέξει μία από τις τρεις επιλογές.

ΠΡΟΒΛΗΜΑΤΑ ΜΕ ΤΑΞΙΝΟΜΗΣΗ ΠΙΝΑΚΑ

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

50. Να γραφεί πρόγραμμα που θα διαβάσει το όνομα και το βαθμό απολυτηρίου 40 φοιτητών. Στη συνέχεια να εμφανίζει τα ονόματα των φοιτητών από τον καλύτερο προς τον χειρότερο.

ΛΥΣΗ

Αφού διαβάσουμε τα ονόματα και τους βαθμούς των φοιτητών σε δύο παράλληλους μονοδιάστατους πίνακες 40 θέσεων, θα πρέπει να κάνουμε φθίνουσα ταξινόμηση των στοιχείων του πίνακα με τους βαθμούς. Εδώ όμως προκύπτει το εξής πρόβλημα: Αν για παράδειγμα είχαμε 5 φοιτητές και είχαμε τις επόμενες τιμές στους δύο πίνακες,

ΟΝ		ΒΑΘΜΟΣ	
1	Γιάννης	1	5,7
2	Θανάσης	2	8,6
3	Μαρία	3	6,3
4	Δημήτρης	4	7,4
5	Ελένη	5	9,1

τότε μετά την ταξινόμηση του πίνακα ΒΑΘΜΟΣ οι πίνακες θα γίνονταν:

ΟΝ		ΒΑΘΜΟΣ	
1	Γιάννης	1	9,1
2	Θανάσης	2	8,6
3	Μαρία	3	7,4
4	Δημήτρης	4	6,3
5	Ελένη	5	5,7

Δηλαδή βλέπουμε ότι ο Γιάννης για παράδειγμα έχει βαθμό 9,1 και όχι 5,7 που είναι ο αρχικός του βαθμός. Για να μην καταστρέψουμε λοιπόν την παραλληλία των δύο πινάκων θα πρέπει κάθε φορά που κάνουμε αντιμετάθεση δύο στοιχείων του πίνακα ΒΑΘΜΟΣ να κάνουμε και αντιμετάθεση των αντίστοιχων στοιχείων του πίνακα ΟΝ. Κάτι τέτοιο ονομάζεται **παράλληλη ταξινόμηση**. Το πρόγραμμα θα έχει ως εξής:

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i,j

ΠΡΑΓΜΑΤΙΚΕΣ: ΒΑΘΜΟΣ[40],TEMP

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[40],TEMP1

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 40

ΔΙΑΒΑΣΕ ΟΝ[i], ΒΑΘΜΟΣ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 40

ΓΙΑ j **ΑΠΟ** 40 **ΜΕΧΡΙ** i **ΜΕ_ΒΗΜΑ** -1

ΑΝ ΒΑΘΜΟΣ[j-1]<ΒΑΘΜΟΣ[j] **ΤΟΤΕ**

TEMP← ΒΑΘΜΟΣ[j-1]

ΒΑΘΜΟΣ[j-1]← ΒΑΘΜΟΣ[j]

ΒΑΘΜΟΣ[j]←TEMP

} αντιμετάθεση των βαθμών

TEMP1←ΟΝ[j-1]

ΟΝ[j-1]←ΟΝ[j]

ΟΝ[j]←TEMP1

} παράλληλη αντιμετάθεση των ονομάτων

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 40

ΓΡΑΨΕ ΟΝ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

51. Ένα περιβαλλοντικό εργαστήριο έκανε μετρήσεις του όζοντος για 30 πόλεις της Ελλάδας καθημερινά για 60 μέρες. Να γραφεί αλγόριθμος που θα διαβάσει τα ονόματα των 30 πόλεων και τις αντίστοιχες μετρήσεις για κάθε μία από τις 60 μέρες. Ο αλγόριθμος να υπολογίζει για κάθε πόλη τη μέση τιμή του όζοντος, και να εμφανίζει τα ονόματα των 30 πόλεων από την πιο καθαρή

ΛΥΣΗ

Οι πίνακες που θα χρησιμοποιήσουμε φαίνονται παρακάτω:

Αφού διαβάσουμε τους πίνακες ON και METP και υπολογίσουμε τον πίνακα MO με τους μέσους όρους, θα πρέπει να κάνουμε αύξουσα ταξινόμηση (από την μικρότερη στη μεγαλύτερη τιμή) του πίνακα MO και παράλληλη αντιμετάθεση του πίνακα ON. Η ιδιαιτερότητα που έχουμε εδώ όμως σε σχέση με την προηγούμενη άσκηση είναι, ότι στην περίπτωση που δύο τιμές του πίνακα MO είναι ίδιες θα πρέπει να ελέξουμε τις αντίστοιχες τιμές του πίνακα ON με τα ονόματα των πόλεων. Αν τα ονόματα των πόλεων δεν τηρούν την αλφαβητική σειρά (δηλαδή αν ισχύει ότι $ON[j-1] > ON[j]$), θα πρέπει να τα αντιμεταθέσουμε. Οι εντολές δηλαδή που θα γράψουμε στην ταξινόμηση θα είναι:

-42-

52. Να γίνει αλγόριθμος που θα δέχεται για κάθε μία από τις 16 ομάδες μπάσκετ ενός πρωταθλήματος το όνομα της ομάδας, τους πόντους που πέτυχαν οι 12 παίκτες τους στο πρωτάθλημα, καθώς και τα ονόματα των παικτών τους. Στη συνέχεια, να διαβάζει το όνομα μιας ομάδας και αν αυτό υπάρχει, να εμφανίζει τα ονόματα των παικτών της από αυτόν που σημείωσε τους περισσότερους πόντους σε αυτόν που σημείωσε τους λιγότερους. Αν η συγκεκριμένη ομάδα δεν υπάρχει να εμφανίζει κατάλληλο μήνυμα.

ΛΥΣΗ

Θα χρησιμοποιήσουμε τους παρακάτω πίνακες:

	1	2	...	16
ON_O				
ON_Π	1	2	...	16
1				
2				
...				
12				
ΠONT	1	2	...	16
1				
2				
...				
12				

Αφού διαβάσουμε τους πίνακες, θα πρέπει να διαβάσουμε το όνομα της ομάδας που θα δώσει ο χρήστης. Στη συνέχεια, να κάνουμε αναζήτηση του ονόματος αυτού στον πίνακα ON_O που θα περιέχει τα ονόματα των ομάδων, και αν το όνομα βρεθεί στη θέση για παράδειγμα Θ του πίνακα, να ταξινομήσουμε τα στοιχεία της Θ στήλης του πίνακα ΠONT με τους πόντους των παικτών, με παράλληλη αντιμετάθεση των στοιχείων της αντίστοιχης στήλης του πίνακα ON_Π με τα ονόματα των παικτών. Για να ταξινομήσουμε τώρα τη στήλη Θ του πίνακα ΠONT με παράλληλη αντιμετάθεση των στοιχείων της στήλης Θ του πίνακα ON_Π, θεωρούμε ότι πρόκειται για δύο μονοδιάστατους πίνακες που έχουν σταθερό το δείκτη της στήλης που βρίσκονται (=Θ) και μεταβάλλεται η γραμμή στην οποία βρίσκονται. Οι εντολές θα είναι:

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 12

ΓΙΑ j ΑΠΟ 12 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ ΠONT[j-1, Θ]<ΠONT[j, Θ] ΤΟΤΕ !φθίνουσα ταξινόμηση (από την μεγαλύτερη στη μικρότερη τιμή)

ΑΝΤΙΜΕΤΑΘΕΣΕ ΠONT[j-1, Θ], ΠONT[j, Θ]

ΑΝΤΙΜΕΤΑΘΕΣΕ ON_Π[j-1, Θ], ON_Π[j, Θ]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Και ολόκληρος ο αλγόριθμος:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 16

ΔΙΑΒΑΣΕ ON_O[j]

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12

ΔΙΑΒΑΣΕ ON_Π[i, j], ΠONT[i, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ONOMA !διάβασμα του ονόματος της ομάδας προς αναζήτηση

i ← 1

BPEΘΗΚΕ ← ΨΕΥΔΗΣ

Θ ← 0

ΟΣΟ i ≤ 12 **ΚΑΙ** BPEΘΗΚΕ = ΨΕΥΔΗΣ **ΕΠΑΝΑΛΑΒΕ**

ΑΝ ON_O[i] = ONOMA **ΤΟΤΕ**

BPEΘΗΚΕ ← ΑΛΗΘΗΣ

Θ ← i

ΑΛΛΙΩΣ

i ← i + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

διάβασμα πινάκων

σειριακή αναζήτηση του ONOMA στον πίνακα ON_O

```

ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ 'Δεν βρέθηκε η ομάδα που δώσατε'
ΑΛΛΙΩΣ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 12
    ΓΙΑ j ΑΠΟ 12 ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1
        ΑΝ ΠΟΝΤ[j-1, Θ]<ΠΟΝΤ[j, Θ] ΤΟΤΕ
            ΑΝΤΙΜΕΤΑΘΕΣΕ ΠΟΝΤ[j-1, Θ], ΠΟΝΤ[j, Θ]
            ΑΝΤΙΜΕΤΑΘΕΣΕ ΟΝ_Π[j-1, Θ], ΟΝ_Π[j, Θ]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
    ΕΜΦΑΝΙΣΕ ΟΝ_Π[i, Θ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΑΝ

```

αν δεν βρέθηκε το όνομα της ομάδας, εμφάνισε μήνυμα, αλλιώς ταξινόμησε κατά φθίνουσα σειρά τα στοιχεία της στήλης Θ στον πίνακα ΠΟΝΤ και αντιμετάθεσε παράλληλα την στήλη Θ του πίνακα ΟΝ_Π. Μετά εμφάνισε τα στοιχεία της στήλης Θ του πίνακα ΟΝ_Π (που είναι πλέον ταξινομημένα από τους περισσότερους πόντους στους λιγότερους).

ΤΕΛΟΣ Άσκηση

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

53. Να γράψετε αλγόριθμο ο οποίος για δεδομένο έναν πίνακα A διαστάσεων N x M θα εμφανίζει τα στοιχεία κάθε στήλης του σε φθίνουσα σειρά.

54. Ο αλγόριθμος της φυσαλίδας όπως διατυπώθηκε στα προηγούμενα παραδείγματα έχει το μειονέκτημα ότι δεν είναι αρκετά "έξυπνος" ώστε να διαπιστώνει στην αρχή ή στο μέσο της διαδικασίας αν ο πίνακας είναι ταξινομημένος. Να σχεδιάσετε μια παραλλαγή του αλγόριθμου αυτού που να σταματά όταν διαπιστωθεί ότι τα στοιχεία του πίνακα είναι ήδη ταξινομημένα.

Υπόδειξη: Να χρησιμοποιήσετε μια βοηθητική μεταβλητή που να ελέγχει το τέλος κάθε επανάληψης του εξωτερικού βρόχου ("για i από 2 μέχρι η") αν για την τρέχουσα τιμή του i έγιναν αντιμεταθέσεις στοιχείων.

Δραστηριότητα ΔΤ2, Κεφάλαιο 3 Τετραδίου μαθητή

55. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει για 50 πόλεις το όνομα τους και τη μέση θερμοκρασία τους που είχαν την τελευταία χρονιά. Ο αλγόριθμος να εμφανίζει τις πόλεις που είχαν αρνητική μέση θερμοκρασία από την πιο ζεστή στην πιο κρύα. Αν δεν υπάρχει πόλη με αρνητική θερμοκρασία, να εμφανίζει κατάλληλο μήνυμα.

56. Στην αρχή της ποδοσφαιρικής περιόδου οι 22 παίκτες μιας ομάδας, οι οποίοι αριθμούνται από 1 έως 22, ψηφίζουν για τους 3 αρχηγούς που θα τους εκπροσωπούν. Κάθε παίκτης μπορεί να ψηφίσει όσους συμπαίκτες του θέλει, ακόμα και τον εαυτό του. Τα αποτελέσματα της ψηφοφορίας καταχωρίζονται σε έναν πίνακα ΨΗΦΟΣ με 22 γραμμές και 22 στήλες, έτσι ώστε το στοιχείο ΨΗΦΟΣ[i,j] να έχει την τιμή 1, όταν ο παίκτης με αριθμό i έχει ψηφίσει τον παίκτη με αριθμό j, και τιμή 0 στην αντίθετη περίπτωση.

Να γράψετε αλγόριθμο ο οποίος:

Δ1. Να διαβάζει τα στοιχεία του πίνακα ΨΗΦΟΣ και να ελέγχει την ορθότητά τους με αποδεκτές τιμές 0 ή 1.

Δ2. Να εμφανίζει το πλήθος των παικτών που δεν ψήφισαν κανέναν.

Δ3. Να εμφανίζει το πλήθος των παικτών που ψήφισαν τον εαυτό τους.

Δ4. Να βρίσκει τους 3 παίκτες που έλαβαν τις περισσότερες ψήφους και να εμφανίζει τους αριθμούς τους και τις ψήφους που έλαβαν. Θεωρήστε ότι δεν υπάρχουν ισοψηφίες. (εξετάσεις 2011)

57. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει το όνομα 10 ευρωπαϊκών πόλεων και το όνομα 10 ασιατικών. Στη συνέχεια για κάθε ευρωπαϊκή πόλη θα διαβάζει την απόσταση της με κάθε ασιατική πόλη. Ο αλγόριθμος τέλος θα διαβάζει το όνομα μιας πόλης το οποίο δεν γνωρίζουμε αν είναι ευρωπαϊκό ή ασιατικό και στη συνέχεια αν υπάρχει θα εμφανίζει τις 5 κοντινότερες της πόλεις που ανήκουν στην άλλη ήπειρο. Αν το όνομα της πόλης που δόθηκε δεν είναι ένα εκ των 20 ονομάτων, θα εμφανίζει κατάλληλο μήνυμα.

58. Μια πολυκατοικία έχει 7 ορόφους και σε κάθε όροφο υπάρχουν 14 διαμερίσματα. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει για κάθε διαμέρισμα το όνομα του ιδιοκτήτη του καθώς και τα κοινόχρηστα που πρέπει να πληρώσει για τον τελευταίο μήνα. Ο αλγόριθμος θα εμφανίζει:

- α. για κάθε όροφο τους ιδιοκτήτες του ξεκινώντας από αυτόν με τα περισσότερα κοινόχρηστα και καταλήγοντας σε αυτόν με τα λιγότερα.
 β. για κάθε όροφο το όνομα του ιδιοκτήτη που πληρώνει τα περισσότερα στον όροφο, ξεκινώντας όμως από τον όροφο που πληρώνουν οι ιδιοκτήτες του περισσότερα και καταλήγοντας στον όροφο που πληρώνουν οι ιδιοκτήτες του τα λιγότερα.

ΣΥΓΧΩΝΕΥΣΗ ΠΙΝΑΚΩΝ

Η συγχώνευση είναι μία από τις βασικές λειτουργίες των πινάκων. Σκοπός της είναι η δημιουργία ενός ταξινομημένου πίνακα από τα στοιχεία δύο ή περισσότερων ταξινομημένων πινάκων. Έστω για παράδειγμα ότι έχουμε τους παρακάτω ταξινομημένους κατά αύξουσα σειρά πίνακες πέντε και έξι θέσεων:

A	1	5	10	16	19
----------	----------	----------	-----------	-----------	-----------

B	2	3	12	16	20	24
----------	----------	----------	-----------	-----------	-----------	-----------

Ο πίνακας που πρέπει να φτιάξουμε πρέπει να είναι 11 θέσεων και να έχει τα στοιχεία του ταξινομημένα κατά αύξουσα σειρά και φαίνεται παρακάτω:

Γ	1	2	3	5	10	12	16	16	19	20	24
----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Για να φτιάξουμε τον αλγόριθμο της συγχώνευσης θα χρησιμοποιήσουμε τρεις δείκτες, έναν για κάθε πίνακα. Για τους πίνακες A και B θα χρησιμοποιήσουμε τους δείκτες i και j αντίστοιχα, οι οποίοι θα δείχνουν κάθε φορά τη θέση του στοιχείου του πίνακα που θα μεταφερθεί στον πίνακα Γ. Για τον πίνακα Γ θα χρησιμοποιήσουμε τον δείκτη κ, ο οποίος θα δείχνει τη θέση του πίνακα στην οποία θα εισαχθεί το επόμενο στοιχείο. Η αρχική τιμή των τριών δεικτών θα είναι 1, και η αρχική μορφή των τριών πινάκων θα είναι η εξής:

	i=1										
	↓										
A	1	5	10	16	19						
		j=1									
		↓									
B	2	3	12	16	20	24					
	κ=1										
	↓										
Γ											

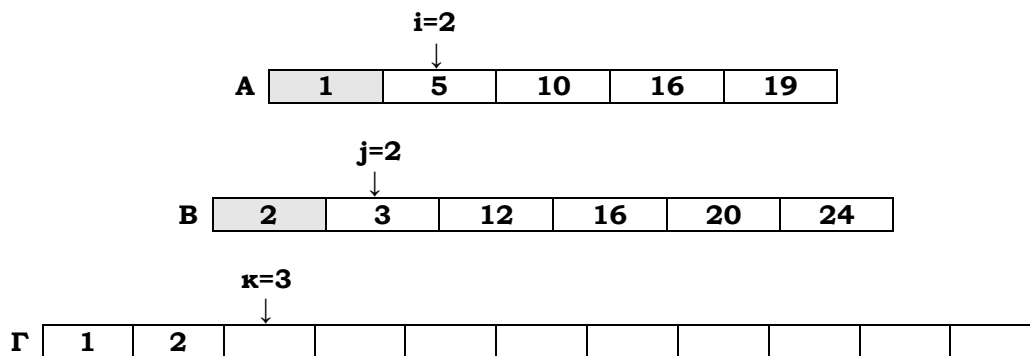
Ο αλγόριθμος της συγχώνευσης χωρίζεται σε δύο φάσεις. Στην πρώτη φάση καταχωρούμε στον πίνακα Γ τα στοιχεία των δύο πινάκων μέχρι να τελειώσουν τα στοιχεία του ενός από τους δύο πίνακες. Στη δεύτερη φάση καταχωρούμε στον πίνακα Γ τα υπόλοιπα στοιχεία του πίνακα αυτού που δεν έχουν εισαχθεί όλα τα στοιχεία του κατά την 1^η φάση. Ας δούμε τα βήματα αναλυτικά:

A' ΦΑΣΗ

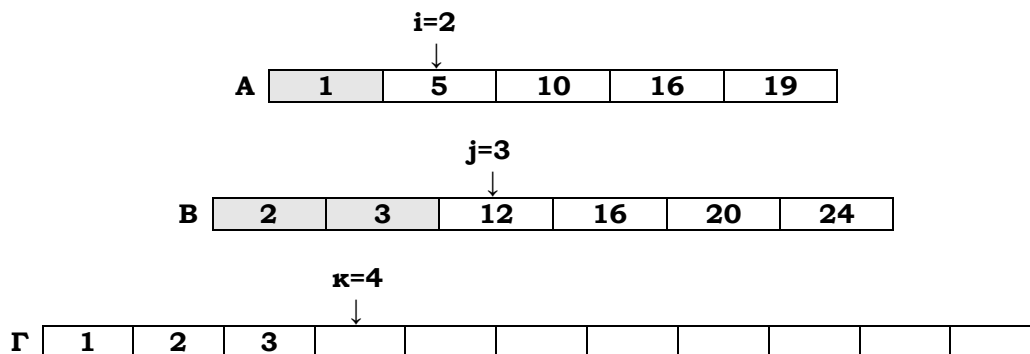
1^ο Βήμα: Συγκρίνουμε τα στοιχεία A[1] και B[1]. Επειδή ισχύει A[1]<B[1], οδηγούμε το A[1] στην πρώτη θέση του πίνακα Γ. Επίσης, το i αυξάνεται κατά 1 για να δείξει το επόμενο στοιχείο του πίνακα A που θα συγκριθεί. Ο δείκτης j, παραμένει ίσος με 1 και ο δείκτης κ γίνεται ίσος με 2, για να δείξει την θέση του πίνακα που θα εισαχθεί το επόμενο στοιχείο. Οι πίνακες θα είναι:

	i=2										
	↓										
A	1	5	10	16	19						
		j=1									
		↓									
B	2	3	12	16	20	24					
	κ=2										
	↓										
Γ	1										

2^ο Βήμα: Συγκρίνονται τα A[2] με το B[1]. Επειδή ισχύει B[1]<A[2], το B[1] εκχωρείται στη 2^η θέση του πίνακα Γ. Τώρα το j θα γίνει 2, το i θα παραμείνει 2 και το κ θα γίνει 3:

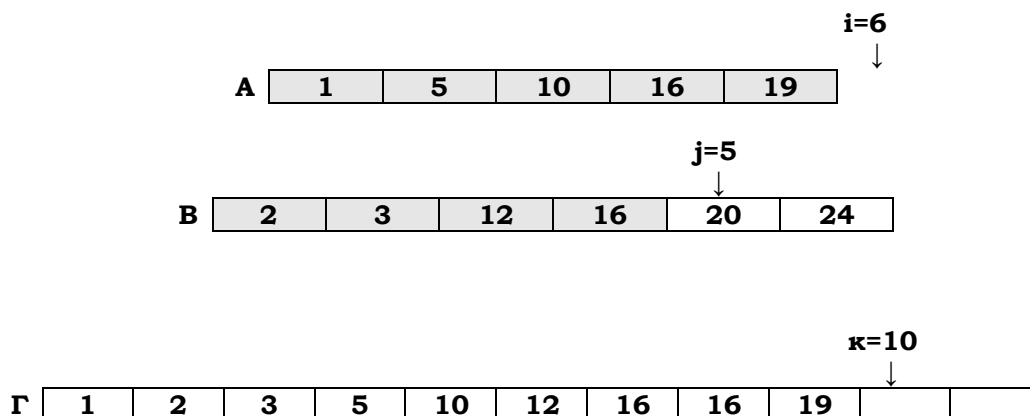


3^ο Βήμα:



.....

9^ο Βήμα:



Στο 9^ο και τελευταίο βήμα της πρώτης φάσης έχουν καταχωριστεί όλα τα στοιχεία του πίνακα A στον πίνακα Γ. Αυτό το καταλαβαίνουμε γιατί η τιμή του δείκτη i είναι 6, δηλαδή μεγαλύτερη από το πλήθος των στοιχείων του A.

Β' ΦΑΣΗ

Στη φάση αυτή θα πρέπει τα υπόλοιπα στοιχεία του πίνακα B να περαστούν με την σειρά που βρίσκονται στις τελευταίες θέσεις του πίνακα Γ. Συνολικά ο αλγόριθμος της συγχώνευσης δύο πινάκων A και B, N και M θέσεων αντίστοιχα έχει ως εξής:

ΑΛΓΟΡΙΘΜΟΣ Συγχώνευση
ΔΕΔΟΜΕΝΑ //A,B,N,M//

i ← 1
j ← 1
κ ← 1

ΟΣΟ (i ≤ N) ΚΑΙ (j ≤ M) **ΕΠΑΝΑΛΑΒΕ** {όσο δεν τελειώσαν τα στοιχεία του A ή του B συνέχισε

ΑΝ A[i] < B[j] **ΤΟΤΕ**

Γ[κ] ← A[i]

i ← i + 1

ΑΛΛΙΩΣ

Γ[κ] ← B[j]

j ← j + 1

ΤΕΛΟΣ_ΑΝ

κ ← κ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

α' φάση

ΑΝ i > N **ΤΟΤΕ**

ΓΙΑ λ **ΑΠΟ** κ **ΜΕΧΡΙ** N + M

Γ[λ] ← B[j]

j ← j + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΛΙΩΣ

ΓΙΑ λ **ΑΠΟ** κ **ΜΕΧΡΙ** N + M

Γ[λ] ← A[i]

i ← i + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΑΝ

ΑΠΟΤΕΛΕΣΜΑΤΑ // Γ //

ΤΕΛΟΣ Συγχώνευση

αν έχουν τελειώσει τα στοιχεία του A, τότε βάλε τα υπόλοιπα στοιχεία του B στον Γ, αλλιώς βάλε τα υπόλοιπα στοιχεία του A στον Γ

β' φάση

Σημείωση: Ένας άλλος αλγόριθμος συγχώνευσης δύο πινάκων N και M στοιχείων, αλλά πολύ πιο αργός σε σχέση με τον προηγούμενο είναι ο εξής: Δημιουργούμε έναν πίνακα Γ, N + M στοιχείων που στις πρώτες N θέσεις του έχει τα στοιχεία του A και στις επόμενες τα στοιχεία του B. Στη συνέχεια, ταξινομούμε τον πίνακα αυτόν. Οι εντολές είναι:

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** N

Γ[i] ← A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** M

Γ[j + N] ← B[j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

γέμισμα του Γ με τους πίνακες A και B

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** N + M

ΓΙΑ j **ΑΠΟ** N + M **ΜΕΧΡΙ** i **ΜΕ_ΒΗΜΑ** - 1

ΑΝ Γ[j - 1] > Γ[j] **ΤΟΤΕ**

ΑΝΤΙΜΕΤΑΘΕΣΕ Γ[j - 1], Γ[j]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ταξινόμηση του Γ

ΑΚΗΣΕΙΣ ΜΕ ΣΥΓΧΩΝΕΥΣΗ ΠΙΝΑΚΩΝ

59. Η Τρίτη Λυκείου ενός σχολείου αποτελείται από 2 τμήματα 25 και 30 μαθητών αντίστοιχα. Να γράψετε αλγόριθμο ο οποίος να διαβάζει για κάθε τμήμα το όνομα των μαθητών του και τον βαθμό απολυτηρίων τους και να τα καταχωρίζει σε μονοδιάστατους πίνακες *ΌνομαA* και *ΌνομαB* για τα ονόματα του α' και β' τμήματος και *ΒαθμόςA* και *ΒαθμόςB* για τους βαθμούς των μαθητών του πρώτου και δεύτερου τμήματος αντίστοιχα. Αν θεωρήσουμε ότι τα στοιχεία των μαθητών κάθε τάξης δόθηκαν από τον καλύτερο μαθητή στον χειρότερο, ο αλγόριθμος να εμφανίζει τους δέκα καλύτερους μαθητές και από τα δύο τμήματα.

60. Δύο εταιρείες τηλεπικοινωνιών έπειτα από τρία χρόνια λειτουργίας αποφάσισαν για λόγους ανταγωνισμού στην παγκόσμια αγορά να συγχωνευτούν ώστε να μειώσουν τα έξοδα τους και να αυξήσουν τα κέρδη τους. Οι δύο εταιρείες πριν από τη συγχώνευση τους απασχολούσαν 20 και 30 άτομα αντίστοιχα. Έπειτα από τη συγχώνευση αποφασίστηκε να κρατήσουν τους 5 πιο ακριβοπληρωμένους και από τις δύο εταιρείες καθώς και τους 15 με τον μικρότερο μισθό. Να υλοποιήσετε πρόγραμμα το οποίο:

- i. θα διαβάζει τον μισθό των 20 υπαλλήλων της πρώτης εταιρείας και θα τους καταχωρίζει σε μονοδιάστατο πίνακα.
- ii. θα διαβάζει τον μισθό των 30 υπαλλήλων της δεύτερης εταιρείας και θα τους καταχωρίζει σε μονοδιάστατο πίνακα.
- iii. θα συγκλινεί τους παραπάνω πίνακες ώστε να προκύψει ένας πίνακας με τους μισθούς των υπαλλήλων.
- iv. αν θεωρήσουμε ότι κατά τη συγκλινεση δεν άλλαξε ο μισθός των υπαλλήλων, θα εμφανίζει πόσα χρήματα γλιτώνουν οι εταιρείες από την πληρωμή μισθών. Παρατήρηση: Θεωρήστε ότι μετά την εισαγωγή των μισθών στα ερωτήματα (i) και (ii) τα στοιχεία των δύο πινάκων είναι ταξινομημένα σε φθίνουσα σειρά.

ΑΣΚΗΣΕΙΣ ΣΥΧΝΟΤΗΤΩΝ

61. Να γραφεί αλγόριθμος ο οποίος:

- i) Θα δέχεται το νούμερο για κάθε μία από τις 5.000 ρίψεις ενός ζαριού (1-6 χωρίς έλεγχο εγκυρότητας) και θα το καταχωρεί σε πίνακα 5.000 θέσεων.
- ii) Θα εμφανίζει τα νούμερα του ζαριού από αυτό με τη μεγαλύτερη συχνότητα εμφάνισης σε αυτό με τη μικρότερη.

ΛΥΣΗ

Αφού διαβάσουμε τον πίνακα με τις ρίψεις των ζαριών (πίνακας P[5000]), θα δημιουργήσουμε έναν πίνακα 6 θέσεων (πίνακας Σ[6]), όπου σε κάθε θέση του θα καταχωρήσουμε την συχνότητα εμφάνισης του νούμερου του ζαριού της αντίστοιχης θέσης. Για παράδειγμα, στη θέση 3 του πίνακα θα έχουμε τη συχνότητα εμφάνισης του 3, δηλαδή πόσες φορές εμφανίστηκε το 3 στις 5.000 ρίψεις του ζαριού (δηλαδή στον μονοδιάστατο πίνακα 5.000 θέσεων που διαβάσαμε στην αρχή). Στην ουσία, κάθε θέση του πίνακα θα είναι ένας μετρητής, άρα πριν από όλα θα πρέπει να τον μηδενίσουμε. Κατόπιν, θα προσπελάσουμε μία μία όλες τις θέσεις του P με τις ρίψεις των ζαριών και θα αυξάνουμε κατά 1 την αντίστοιχη θέση του πίνακα Σ. Οι εντολές θα είναι:

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6 Σ[i] ← 0 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	}	μηδενισμός του πίνακα συχνοτήτων
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5000 K ← P[i] Σ[K] ← Σ[K] + 1 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	}	αν για παράδειγμα η θέση i του πίνακα P έχει τιμή 5, τότε αυξάνεται το Σ[5] κατά 1

Αφού δημιουργήσουμε τον πίνακα Σ, θα πρέπει να τον ταξινομήσουμε κατά φθίνουσα σειρά (από τη μεγαλύτερη στη μικρότερη τιμή). Αν για παράδειγμα ο πίνακας Σ έχει τις παρακάτω τιμές:

Σ	400	600	1500	1000	300	1700
	1	2	3	4	5	6

Μετά την ταξινόμηση θα είναι:

Σ	1700	1500	1000	600	400	300
	1	2	3	4	5	6

Παρατηρούμε ότι έχει χαθεί η αντιστοιχία νούμερου- συχνότητας εμφάνισης, δηλαδή ενώ το νούμερο 6 είχε τη μεγαλύτερη συχνότητα εμφάνισης, τώρα στη θέση 6 έχουμε την μικρότερη συχνότητα εμφάνισης. Για να μη συμβεί αυτό, πρέπει να φτιάξουμε έναν πίνακα 6 θέσεων παράλληλο με τον πίνακα Σ (πίνακας Δ[6]), που κάθε θέση του θα είναι το νούμερο του ζαριού που αντιστοιχεί στην συγκεκριμένη θέση. Ο πίνακας αυτός καλείται **πίνακας δεικτών**. Οι δύο πίνακες πριν την ταξινόμηση θα είναι:

Σ	400	600	1500	1000	300	1700
	1	2	3	4	5	6

Δ	1	2	3	4	5	6
	1	2	3	4	5	6

Κατά την ταξινόμηση του πίνακα Σ, θα πρέπει να γίνει και ταυτόχρονη αντιμετάθεση των στοιχείων του Δ, για να μη χαθεί η αντιστοιχία νούμερου-συχνότητας εμφάνισης. Οι πίνακες μετά την ταξινόμηση θα είναι:

Σ	1700	1500	1000	600	400	300
	1	2	3	4	5	6

Δ	6	3	4	2	1	5
----------	---	---	---	---	---	---

Ακολουθεί ο αλγόριθμος συνολικά:

ΑΛΓΟΡΙΘΜΟΣ Άσκηση

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5000

ΔΙΑΒΑΣΕ P[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} διάβασμα των ρίψεων του ζαριού

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6

$\Sigma[i] \leftarrow 0$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} μηδενισμός του πίνακα συχνοτήτων

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5000

$K \leftarrow P[i]$

$\Sigma[K] \leftarrow \Sigma[K] + 1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} υπολογισμός του πίνακα συχνοτήτων

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6

$\Delta[i] \leftarrow i$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} φτιάχνουμε τον πίνακα δεικτών

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 6

ΓΙΑ j ΑΠΟ 6 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ $\Sigma[j-1] < \Sigma[j]$ ΤΟΤΕ

ΑΝΤΙΜΕΤΑΘΕΣΕ $\Sigma[j-1]$, $\Sigma[j]$

ΑΝΤΙΜΕΤΑΘΕΣΕ $\Delta[j-1]$, $\Delta[j]$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} ταξινόμηση του πίνακα συχνοτήτων και παράλληλη αντιμετάθεση του πίνακα δεικτών

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6

ΕΜΦΑΝΙΣΕ $\Delta[i]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

} εμφάνιση του πίνακα δεικτών

ΤΕΛΟΣ Άσκηση

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

62. Σε μια τάξη 25 μαθητών ο καθηγητής της Χημείας έδωσε στους μαθητές του 30 χημικές ενώσεις και οι μαθητές έπρεπε να βρουν τις ονομασίες τους. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει για κάθε μαθητή το όνομα του καθώς και για καθεμία από τις 30 χημικές ενώσεις τη λέξη «Επιτυχία», αν βρήκε την ονομασία, ή την λέξη «Αποτυχία», αν η ονομασία που έδωσε είναι άστοχη. Ο αλγόριθμος να εμφανίζει:

α. για κάθε μαθητή το όνομα του και το πλήθος των ονομασιών που βρήκε, β. να εμφανίζει μήνυμα της μορφής «οι περισσότεροι μαθητές βρήκαν το όνομα ... χημικών ενώσεων», όπου το κενό θα συμπληρωθεί με το πλήθος των σωστών χημικών ενώσεων που ήταν το πιο συχνό από τους μαθητές.

63. Το Υπουργείο Υγείας πραγματοποιεί μια έρευνα για τις επιλογές των τελειοφοίτων Ιατρικής όσον αφορά την επιλογή ειδικότητας.

Να αναπτύξετε αλγόριθμο, ο οποίος:

α. θα διαβάζει τα ονόματα των 20 ειδικοτήτων σε πίνακα ΕΙΔ[20].

β. για κάθε τελειόφοιτο Ιατρικής:

1. θα διαβάζει επαναληπτικά την ειδικότητα που προτίθεται να επιλέξει.

2. θα εντοπίζει σε ποια θέση του πίνακα ΕΙΔ αντιστοιχεί η συγκεκριμένη ειδικότητα. Αν δεν αντιστοιχεί σε καμία ειδικότητα θα εμφανίζεται κατάλληλα διαμορφωμένο μήνυμα.

3. θα καταμετρά την επιλογή του, χρησιμοποιώντας κατάλληλο πίνακα μετρητών Π[20].

Η παραπάνω επαναληπτική διαδικασία ολοκληρώνεται μόλις δοθεί ως ειδικότητα, η λέξη "τέλος".

γ. θα εμφανίζει τις ειδικότητες που έχουν το δεύτερο μεγαλύτερο αριθμό προτιμήσεων. Στην ακραία περίπτωση που όλες οι ειδικότητες έχουν τον ίδιο αριθμό προτιμήσεων, να εμφανίζεται κατάλληλα διαμορφωμένο μήνυμα.

Παρατήρηση: μπορεί γενικά να υπάρχουν περισσότερες ειδικότητες με ίδιο αριθμό προτιμήσεων.

64. Ένα παιχνίδι ΛΟΤΤΟ έχει 49 αριθμούς (από το 1 έως και το 49) και σε κάθε κλήρωση κληρώνονται 6 από αυτούς. Να γράψετε αλγόριθμο ο οποίος θα διαβάζει τους αριθμούς του ΛΟΤΤΟ που κληρώθηκαν τις

τελευταίες 500 φορές και θα εμφανίζει τα 49 νούμερα αρχίζοντας από αυτό που κληρώθηκε τις περισσότερες φορές και καταλήγοντας σε αυτό που κληρώθηκε τις λιγότερες.

ΛΥΜΕΝΑ ΘΕΜΑΤΑ ΕΞΕΤΑΣΕΩΝ

65. Σ' έναν διαγωνισμό συμμετέχουν 100 υποψήφιοι. Κάθε υποψήφιος διαγωνίζεται σε 50 ερωτήσεις πολλαπλής επιλογής. Να αναπτυχθεί αλγόριθμος που να κάνει τα παρακάτω:

α. Να καταχωρίζει σε πίνακα ΑΠ[100, 50] τα αποτελέσματα των απαντήσεων του κάθε υποψηφίου σε κάθε ερώτηση. Κάθε καταχώριση μπορεί να είναι μόνο μία από τις παρακάτω:

- i. Σ, αν είναι σωστή η απάντηση,
- ii. Λ, αν είναι λανθασμένη η απάντηση και
- iii. Ξ, αν ο υποψήφιος δεν απάντησε.

Να γίνεται έλεγχος των δεδομένων εισόδου.

β. Να βρίσκει και να τυπώνει τους αριθμούς των ερωτήσεων που παρουσιάζουν τον μεγαλύτερο βαθμό δυσκολίας, δηλαδή έχουν το μικρότερο πλήθος σωστών απαντήσεων.

γ. Αν κάθε Σ βαθμολογείται με 2 μονάδες, κάθε Λ με -1 μονάδα και κάθε Ξ με 0 μονάδες, τότε:

- i. να δημιουργεί έναν μονοδιάστατο πίνακα ΒΑΘ[100], κάθε στοιχείο του οποίου θα περιέχει αντίστοιχα τη συνολική βαθμολογία ενός υποψηφίου.
- ii. να τυπώνει το πλήθος των υποψηφίων που συγκέντρωσαν βαθμολογία μεγαλύτερη από 50.

(Εξετάσεις 2005)

ΛΥΣΗ

α' ερώτημα: Θα διαβάσουμε τον πίνακα ΑΠ κάνοντας έλεγχο εγκυρότητας για τα δεδομένα εισόδου. Κάθε τιμή που διαβάζουμε μπορεί να είναι μόνο 'Σ' ή 'Λ' ή 'Ξ':

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 50
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΔΙΑΒΑΣΕ ΑΠ[i, j]
      ΜΕΧΡΙΣ_ΟΤΟΥ ΑΠ[i, j]='Σ' Η' ΑΠ[i, j]='Λ' Η' ΑΠ[i, j]='Ξ'
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

β' ερώτημα: Για να βρούμε το πλήθος σωστών απαντήσεων για κάθε ερώτηση θα φτιάξουμε έναν πίνακα 50 θέσεων, παράλληλο με τις στήλες του πίνακα ΑΠ:

ΑΠ	1	2	...	50
1				
2				
...				
100				

	↓	↓	↓	↓
Π				
	1	2	...	50

Στη συνέχεια, πρέπει να βρούμε το ελάχιστο στοιχείο του πίνακα αυτού, και να προσπελάσουμε όλα τα στοιχεία του πίνακα, έτσι ώστε αν κάποιο στοιχείο ισούται με το ελάχιστο, να εμφανίζουμε τη θέση του που στην ουσία είναι ο αριθμός της ερώτησης. Οι εντολές για όλα αυτά θα είναι:

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 50
  Π[j] ← 0
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΑΝ ΑΠ[i, j]='Σ' ΤΟΤΕ
      Π[j] ← Π[j] + 1
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  min ← Π[1]
  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 50
    ΑΝ Π[i] < min ΤΟΤΕ
      min ← Π[i]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    ΑΝ Π[i] = min ΤΟΤΕ
      ΕΜΦΑΝΙΣΕ i
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

υπολογισμός του πίνακα Π

υπολογισμός του ελάχιστου στοιχείου

εμφάνιση των θέσεων του πίνακα Π που έχουν τιμή ίση με το min

γ' ερώτημα: Για να φτιάξουμε τον πίνακα ΒΑΘ, με τις βαθμολογίες των υποψηφίων, θα προσπελάσουμε τον πίνακα Π κατά γραμμές και θα ελέγχουμε ένα-ένα όλα τα στοιχεία ως εξής: αν το στοιχείο είναι 'Σ' θα αυξάνουμε το αντίστοιχο στοιχείο του πίνακα ΒΑΘ κατά 2, αν είναι 'Λ' θα προσθέτουμε το -1, ενώ αν είναι 'Ξ' δεν θα κάνουμε τίποτα. Στη συνέχεια θα μετρήσουμε πόσα από τα στοιχεία του πίνακα ΒΑΘ είναι μεγαλύτερα από 50:

ΑΠ	1	2		50		ΒΑΘ
1					→	1
2					→	2
....						
100					→	100

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΒΑΘ[i]←0

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 50

ΑΝ ΑΠ[i, j]='Σ' ΤΟΤΕ

ΒΑΘ[i]←ΒΑΘ[i]+2

ΑΛΛΙΩΣ_ΑΝ ΑΠ[i, j]='Λ' ΤΟΤΕ

ΒΑΘ[i]←ΒΑΘ[i]-1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

υπολογισμός του πίνακα ΒΑΘ

ΠΛΗΘΟΣ←0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΑΝ ΒΑΘ[i]>50 ΤΟΤΕ

ΠΛΗΘΟΣ←ΠΛΗΘΟΣ+1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΕΜΦΑΝΙΣΕ ΠΛΗΘΟΣ

υπολογισμός του πλήθους με βαθμολογία μεγαλύτερη του 50

66. Για τη διεκδίκηση μιας θέσης υποτροφίας εξετάστηκαν και βαθμολογήθηκαν πενήντα (50) υποψήφιοι σε τρία μαθήματα. Ο υπολογισμός του τελικού βαθμού κάθε υποψηφίου γίνεται ως εξής: Αν ο βαθμός του σε κάποιο από τα τρία μαθήματα είναι μικρότερος του 6, τότε ο τελικός βαθμός του είναι μηδέν (0). Διαφορετικά ο βαθμός του 1ου μαθήματος συμμετέχει στον υπολογισμό του τελικού βαθμού με συντελεστή 20%, ο βαθμός του 2ου μαθήματος με συντελεστή 35% και ο βαθμός του 3ου μαθήματος με συντελεστή 45%. Να αναπτυχθεί αλγόριθμος ο οποίος:

- διαβάζει τα ονόματα των 50 υποψηφίων και τα καταχωρίζει σε πίνακα,
- διαβάζει για κάθε υποψήφιο τους βαθμούς του σε καθένα από τα τρία μαθήματα και τους καταχωρίζει σε πίνακα δύο διαστάσεων.
- υπολογίζει τον τελικό βαθμό κάθε υποψηφίου και τον καταχωρίζει σε πίνακα,
- ταξινομεί τα ονόματα και τους τελικούς βαθμούς των υποψηφίων σε φθίνουσα σειρά ως προς τον τελικό βαθμό.
- εμφανίζει για όσους υποψηφίους έχουν τελικό βαθμό μεγαλύτερο του μηδενός (0) το όνομα και τον τελικό βαθμό τους.
- εμφανίζει το ποσοστό των υποψηφίων που έχουν τελικό βαθμό μηδέν (0).

(Εξετάσεις 2006)

ΛΥΣΗ

Οι πίνακες που θα χρησιμοποιηθούν είναι οι εξής:

ONOMA		B	1	2	3		TB
1		1				→	1
2		2				→	2
....						→	
50		50				→	50

ερωτήματα α,β:

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50

ΔΙΑΒΑΣΕ ΟΝΟΜΑ[i]

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 3

ΔΙΑΒΑΣΕ B[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ερώτημα γ:

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
  ΑΝ B[i,1]<6 Η' B[i,2]<6 Η' B[i,3]<6 ΤΟΤΕ
    TB[i]←0
  ΑΛΛΙΩΣ
    TB[i]← B[i,1]*0,2+ B[i,2]*0,35+ B[i,3]*0,45
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ερώτημα δ:

```
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 50
  ΓΙΑ j ΑΠΟ 50 ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1
    ΑΝ TB[j-1]<TB[j] ΤΟΤΕ
      ΑΝΤΙΜΕΤΑΘΕΣΕ TB[j-1],TB[j]
      ΑΝΤΙΜΕΤΑΘΕΣΕ ΟΝΟΜΑ[j-1],ΟΝΟΜΑ[j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ερώτημα ε:

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
  ΑΝ TB[i]>0 ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ ΟΝΟΜΑ[i],TB[i]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ερώτημα στ:

```
πλήθος←0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
  ΑΝ TB[i]=0 ΤΟΤΕ
    πλήθος←πλήθος+1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ποσοστό←(πλήθος/50)*100
ΕΜΦΑΝΙΣΕ ποσοστό
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ

67. Στους προκριματικούς αγώνες ιππικού τριάθλου συμμετέχουν 16 αθλητές. Τα αγωνίσματα είναι: ιππική δεξιотехνία, υπερπηδήση εμποδίων και ελεύθερη ιππασία. Ο κάθε αθλητής βαθμολογείται ξεχωριστά σε καθένα από τα τρία αγωνίσματα. Να σχεδιάσετε αλγόριθμο ο οποίος:

- καταχωρίζει σε πίνακα τις ονομασίες των τριών αγωνισμάτων, όπως αυτές δίνονται παραπάνω.
- διαβάζει για κάθε αθλητή όνομα, επίθετο, όνομα αλόγου με το οποίο αγωνίζεται και τους βαθμούς του σε κάθε αγώνισμα και θα καταχωρίζει τα στοιχεία σε πίνακες,
- διαβάζει το όνομα και το επίθετο ενός αθλητή και θα εμφανίζει το όνομα του αλόγου με το οποίο αγωνίστηκε και τη συνολική του βαθμολογία στα τρία αγωνίσματα. Αν δεν υπάρχει ο αθλητής, θα εμφανίζει κατάλληλα διαμορφωμένο μήνυμα,
- εμφανίζει την ονομασία του αγωνίσματος (ή των αγωνισμάτων) με το μεγαλύτερο «άνοιγμα βαθμολογίας». Ως «άνοιγμα βαθμολογίας» να θεωρήσετε τη διαφορά ανάμεσα στην καλύτερη και στη χειρότερη βαθμολογία του αγωνίσματος.

(Εξετάσεις 2006)

68. Σε έναν πανελλήνιο σχολικό διαγωνισμό μετέχουν 20 σχολεία. Κάθε σχολείο αξιολογεί 5 άλλα σχολεία και δεν αυτοαξιολογείται. Η βαθμολογία κυμαίνεται από 1 έως και 10. Να γράψετε τμήμα αλγόριθμου που να διαβάζει τα ονόματα των σχολείων και να τα αποθηκεύει σε μονοδιάστατο πίνακα A 20 θέσεων, να εισάγει αρχικά την τιμή 0 σε όλες τις θέσεις ενός διοδιάστατου πίνακα B 20 γραμμών και 20 στηλών. Στη συνέχεια να καταχωρίζει στον πίνακα B τη βαθμολογία που δίνει κάθε σχολείο για 5 άλλα σχολεία.

Σημείωση: Στη θέση i, j του πίνακα B αποθηκεύεται ο βαθμός που το σχολείο i δίνει στο σχολείο j, όπως φαίνεται στο παράδειγμα που ακολουθεί.

Ο αλγόριθμος να υπολογίζει τη συνολική βαθμολογία του κάθε σχολείου και να την καταχωρίζει σε μονοδιάστατο πίνακα 20 θέσεων με όνομα SUM και να εμφανίζει τα ονόματα και τη συνολική βαθμολογία όλων των σχολείων κατά φθίνουσα σειρά της συνολικής βαθμολογίας.

Παράδειγμα

	Σχολείο 1	Σχολείο 2	...	Σχολείο 5	...	Σχολείο 18	Σχολείο 19	Σχολείο 20
Σχολείο 1			...					
Σχολείο 2	10			8		4	8	6
...								
Σχολείο 20				4				

Στο ανωτέρω παράδειγμα: Το Σχολείο2 έδωσε την παρακάτω βαθμολογία: στο Σχολείο 1 τον βαθμό 10, στο Σχολείο5 τον βαθμό 8, στο Σχολείο 18 τον βαθμό 4, στο Σχολείο19 τον βαθμό 8, και στο Σχολείο20 τον βαθμό 6. Το Σχολείο5 έχει πάρει την παρακάτω βαθμολογία: από το Σχολείο2 τον βαθμό 8 και από το Σχολείο20 τον βαθμό 4.

(Εξετάσεις 2005)

69. Σε έναν διαγωνισμό κρασιού συμμετέχουν 20 πόλεις. Στον διαγωνισμό αυτό κάθε πόλη μπορεί να βαθμολογήσει τις υπόλοιπες εκτός του εαυτού της. Να γράψετε αλγόριθμο που θα διαβάσει:

- το όνομα κάθε πόλης.
 - τη βαθμολογία που δίνει κάθε πόλη στις υπόλοιπες την οποία να την καταχωρίζει σε πίνακα 20 x 20, όπου η κύρια διαγώνιος της θα περιέχει την τιμή 0 (αφού καμία πόλη δεν μπορεί να βαθμολογήσει τον εαυτό της). Έτσι στον πίνακα αυτό για παράδειγμα στη θέση 2, 3 περιέχεται η βαθμολογία που έβαλε η δεύτερη πόλη στην τρίτη.
- Ο αλγόριθμος θα εμφανίζει:
- την πόλη με τη μεγαλύτερη βαθμολογία.
 - την πόλη που έβαλε τη μεγαλύτερη μέση βαθμολογία.

70. Μια αεροπορική εταιρεία ταξιδεύει σε 15 προορισμούς του εσωτερικού. Στα πλαίσια της οικονομικής πολιτικής που πρόκειται να εφαρμόσει, κατέγραψε το ποσοστό πληρότητας των πτήσεων για κάθε μήνα του προηγούμενου ημερολογιακού έτους. Η πολιτική έχει ως εξής: Δεν θα γίνει καμία περικοπή σε προορισμούς, στους οποίους το μέσο ετήσιο ποσοστό πληρότητας των πτήσεων είναι μεγαλύτερο του 65. Θα γίνουν περικοπές πτήσεων σε προορισμούς, στους οποίους το μέσο ετήσιο ποσοστό πληρότητας των πτήσεων κυμαίνεται από 40 έως και 65. Οι περικοπές θα γίνουν μόνο σε εκείνους τους μήνες που το ποσοστό πληρότητας τους είναι μικρότερο του 40. Θα καταργηθούν οι προορισμοί στους οποίους το μέσο ετήσιο ποσοστό πληρότητας των πτήσεων είναι μικρότερο του 40. Να γράψετε αλγόριθμο ο οποίος:

- να διαβάσει τα ονόματα των 15 προορισμών και να τα αποθηκεύει σε έναν μονοδιάστατο πίνακα.
- να διαβάσει τα ποσοστά πληρότητας των πτήσεων των 15 προορισμών για κάθε μήνα και να τα αποθηκεύει σε διδιάστατο πίνακα κάνοντας έλεγχο στην καταχώριση των δεδομένων, ώστε να καταχωρίζονται μόνο οι τιμές που είναι από 0 έως και 100.
- να βρίσκει και να τυπώνει τα ονόματα των προορισμών που δεν θα γίνει καμία περικοπή πτήσεων.
- να βρίσκει και να τυπώνει τα ονόματα των προορισμών που θα καταργηθούν.
- να βρίσκει και να τυπώνει τα ονόματα των προορισμών στους οποίους θα γίνουν περικοπές πτήσεων, καθώς και τους μήνες (αύξοντα αριθμό μήνα) που θα γίνουν οι περικοπές.

(Εξετάσεις 2005)

71. Για την εθνοφρουρά της Αγγλίας πρέπει να μπουν ακόμα 20 άτομα. Τα άτομα διαλέγονται με βάση το ύψος και πολλές φορές πρέπει να ληφθεί υπόψη και το βάρος τους. Το τέλειο ύψος για την εθνοφρουρά θεωρείται το 1,97 μέτρα και όσο πιο κοντά είναι το ύψος ενός υποψηφίου στο συγκεκριμένο ύψος έχει περισσότερες πιθανότητες να επιλεγεί. Να γράψετε αλγόριθμο ο οποίος θα διαβάσει το όνομα, το ύψος και το βάρος των 110 υποψηφίων για τις 20 διαθέσιμες θέσεις. Ο αλγόριθμος θα εμφανίζει τα 20 άτομα που θα μπουν στην εθνοφρουρά. Σε περίπτωση που κάποιοι υποψήφιοι έχουν το ίδιο ύψος και πρέπει να επιλέξουν κάποιους από αυτούς, τότε θα επιλεγούν αυτοί με το μικρότερο βάρος.

Παρατήρηση: Θεωρήστε ότι όλοι οι υποψήφιοι έχουν μεταξύ τους διαφορετικό ύψος ή βάρος. Επίσης θεωρήστε ότι ο χρήστης θα δώσει έγκυρες τιμές.

72. Σε ένα πρωτάθλημα μπάσκετ συμμετέχουν 15 ομάδες με 10 παίκτες η καθεμία. Να γράψετε πρόγραμμα το οποίο θα δέχεται το όνομα κάθε ομάδας καθώς και τα ονόματα των παικτών της και τους πόντους που έβαλαν σε όλο το πρωτάθλημα. Το πρόγραμμα θα υλοποιεί τα εξής:

- Θα εμφανίζει το όνομα της πιο επιθετικής ομάδας (δηλαδή αυτής που έβαλε τα πιο πολλά καλάθια).

β. Θα εμφανίζει το όνομα του πιο επιθετικού παίκτη του πρωταθλήματος.

γ. Θα δέχεται ένα όνομα παίκτη και, αν υπάρχει, θα εμφανίζει όλους τους πόντους των παικτών της ομάδας του σε φθίνουσα σειρά.

73. Μια εταιρεία έχει καταγράψει για τους 20 υπάλληλους της το όνομα και τις μηνιαίες πωλήσεις του καθενός για το έτος που πέρασε. Παρατηρώντας ο διευθυντής της εταιρείας τα ονόματα και τις μηνιαίες πωλήσεις κάθε υπαλλήλου παρατήρησε ότι οι πληροφορίες κάποιων υπαλλήλων έχουν μπερδευτεί μεταξύ τους. Να γράψετε αλγόριθμο ο οποίος:

α. θα διαβάζει για κάθε υπάλληλο το όνομα του και τις μηνιαίες του πωλήσεις,

β. θα διαβάζει δύο ονόματα υπαλλήλων και θα αντιμεταθέτει τα στοιχεία τους. Η διαδικασία (β) θα επαναλαμβάνεται μέχρι να δοθεί όνομα που δεν υπάρχει στην παραπάνω λίστα,

γ. αφού τα στοιχεία των υπαλλήλων έχουν διορθωθεί, θα εμφανίζει τα ονόματα και τις μηνιαίες τους πωλήσεις.

74. Στο ελληνικό εθνικό οδικό δίκτυο υπάρχουν 28 σταθμοί διοδίων και καταγράφονται για κάθε σταθμό οι ημερήσιες διελεύσεις αυτοκινήτων για τον μήνα Ιανουάριο (31 μέρες), ώστε να βγουν κάποια στατιστικά στοιχεία χρήσιμα για τη νέα τιμή διέλευσης που πρέπει το κράτος να θέσει σε κάθε σταθμό, ώστε να αυξήσει τα κέρδη του. Να γράψετε πρόγραμμα που θα διαβάζει για κάθε σταθμό την τιμή διέλευσης ενός αυτοκινήτου καθώς και την ημερήσια διέλευση αυτοκινήτων για τον μήνα Ιανουάριο. Το πρόγραμμα να εμφανίζει:

α. την ημέρα με τις περισσότερες διελεύσεις αυτοκινήτων,

β. την ημέρα με τα περισσότερα κέρδη του κράτους,

γ. αν υπάρχει μέρα που ο σταθμός με την υψηλότερη τιμή διέλευσης είχε λιγότερα κέρδη από τον σταθμό με τη μικρότερη τιμή διέλευσης.

Παρατήρηση: Για το πρόγραμμα θεωρήστε ότι ο χρήστης δίνει έγκυρες τιμές και για το τρίτο ερώτημα θεωρήστε ότι όλοι οι σταθμοί έχουν διαφορετική τιμή διέλευσης ενός αυτοκινήτου.

75. Μια αλυσίδα καταστημάτων στον χώρο των καλλυντικών έχει 120 προϊόντα. Για κάθε προϊόν καταγράφεται η τιμή του η οποία είναι η ίδια για όλα τα υποκαταστήματα. Αν ο αριθμός των καταστημάτων είναι 9, να γράψετε αλγόριθμο ο οποίος:

α. θα διαβάζει:

1. για κάθε κατάστημα τον δήμο στον οποίο ανήκει, θεωρώντας ότι κανένας δήμος δεν έχει δύο καταστήματα,

2. την τιμή του κάθε προϊόντος,

3. τον αριθμό των διαθέσιμων κομματιών του κάθε προϊόντος σε κάθε υποκατάστημα.

β. θα εμφανίζει:

1. για κάθε υποκατάστημα τη συνολική τιμή όλων των προϊόντων του,

2. το προϊόν που έχει τα περισσότερα κομμάτια σε όλα τα υποκαταστήματα,

3. τον δήμο του υποκαταστήματος το οποίο διαθέτει τα περισσότερα κομμάτια του πιο ακριβού προϊόντος.

76. Στο παιχνίδι «Ιππότες του ονείρου» κάθε παίκτης περνάει από 25 στάδια δοκιμασιών. Στο κάθε στάδιο είναι υποχρεωμένος να περάσει από 6 εμπόδια. Στην περίπτωση που αποτύχει σε περισσότερα από 3 εμπόδια, ακόμα και σε ένα από τα 25 στάδια, τότε αποτυγχάνει να γίνει κανονικός ιππότης. Παρ' όλα αυτά, για να γίνει κάποιος ιππότης, θα πρέπει, εκτός του να περάσει από όλα τα στάδια, να έχει κατά μέσο όρο το πολύ ένα αποτυχημένο εμπόδιο σε καθένα από τα 5 τελευταία στάδια. Να γράψετε πρόγραμμα που για 40 υποψήφιους ιππότες να διαβάζει το όνομα τους και τις επιτυχίες τους σε καθένα από τα 25 εμπόδια και να εμφανίζει τα ονόματα των παικτών που θα χρισθούν ιππότες.

77. Στους τελευταίους Ολυμπιακούς Αγώνες στο άλμα εις μήκος οι αθλητές που θα διεκδικήσουν μια θέση για τον τελικό του αγωνίσματος είναι 27. Στον τελικό προκρίνονται οι 12 καλύτεροι καθώς και αυτοί που η τελική επίδοσή τους είναι μεγαλύτερη ή ίση των 8 μέτρων. Στο άθλημα αυτό κάθε αθλητής έχει 6 προσπάθειες και ως τελική του επίδοση λαμβάνεται η επίδοσή του καλύτερου άλματος. Να γράψετε αλγόριθμο που θα διαβάζει το όνομα των 27 αθλητών καθώς και την επίδοσή τους σε καθεμία από τις 6 προσπάθειες. Να εμφανίζει τα ονόματα που προκρίνονται στον τελικό του αγωνίσματος.

Παρατήρηση: Θεωρήστε ότι όλα τα άλματα είναι έγκυρα και ότι όλοι οι αθλητές έχουν διαφορετική τελική επίδοση.

78. Μια αίθουσα ενός πανεπιστημίου αποτελείται από 20 σειρές καθισμάτων και κάθε σειρά έχει από 10 καθίσματα. Αν για την αναπαράσταση των καθισμάτων χρησιμοποιήσουμε έναν πίνακα 20 x 10 (20 σειρές καθισμάτων x 10 καθίσματα η καθεμία), να γράψετε πρόγραμμα που θα διαβάζει το όνομα των 200 φοιτητών καθώς και τη θέση (σειρά, κάθισμα) που θα τοποθετηθεί ο καθένας. Πριν, όμως, από την εισαγωγή της θέσης που θα τοποθετηθεί κάθε φοιτητής το πρόγραμμα να εμφανίζει τις διαθέσιμες θέσεις που υπάρχουν στον πίνακα. Για παράδειγμα, πριν δοθεί η θέση του πρώτου φοιτητή θα πρέπει να εμφανιστούν

όλες οι θέσεις, ενώ πριν δοθεί η θέση του 200ού φοιτητή θα πρέπει να εμφανιστεί μόνο μία. Θεωρήστε ότι ο χρήστης αφού διαβάσει τις διαθέσιμες θέσεις δεν κάνει λάθος αλλά δίνει τις συνεταγμένες μίας διαθέσιμης θέσης.

Συμβουλή: Δώστε για αρχική τιμή σε κάθε θέση του δισδιάστατου πίνακα το κενό ' '.

79. Ένας διαγωνισμός τραγουδιού στην Ευρώπη διεξάγεται ως εξής: Γίνεται μια πρώτη ακρόαση των τραγουδιών κάθε χώρας από την Κριτική Επιτροπή, η οποία δίνει κάποιους βαθμούς σε κάθε τραγούδι (από 1-100). Έστω ότι είναι γνωστοί οι βαθμοί που δόθηκαν στο τραγούδι κάθε χώρας. Να γράψετε έναν αλγόριθμο που θα επιλέγει για τη συνέχεια στη δεύτερη φάση του διαγωνισμού τις χώρες με τη μεγαλύτερη βαθμολογία κάθε φορά ώστε το άθροισμα της βαθμολογίας όλων των τραγουδιών που θα προχωρήσουν στη φάση να είναι μικρότερη από 1.000 βαθμούς.

Δραστηριότητα ΔΣ5, Κεφάλαιο 4 Τετραδίου Μαθητή

80. Να αναπτύξετε αλγόριθμο που θα δέχεται τον κωδικό που έβαλαν 5.000 άτομα στο κινητό τους. Στη συνέχεια θα εμφανίζει μήνυμα για το αν υπάρχουν κωδικοί που δόθηκαν περισσότερες από 3 φορές.

81. Στις τελευταίες εκλογές του 15μελούς συμβουλίου κάθε μαθητής θα πρέπει να ψηφίσει έναν υποψήφιο, ώστε οι 15 μαθητές που μάζεψαν τις περισσότερες ψήφους να αποτελέσουν τα μέλη του. Αν το σχολείο αποτελείται από 300 μαθητές, να γράψετε αλγόριθμο ο οποίος αρχικά:

α. θα διαβάσει το όνομα κάθε μαθητή (θεωρήστε τα όλα διαφορετικά μεταξύ τους),
β. θα διαβάσει για κάθε μαθητή την τιμή «ΝΑΙ» ή την τιμή «ΟΧΙ», αν θέλει ή όχι να βάλει υποψηφιότητα (δεν χρειάζεται έλεγχος εγκυρότητας),

Στη συνέχεια ο αλγόριθμος θα επιτρέπει σε κάθε μαθητή να ψηφίσει έναν υποψήφιο, δίνοντας το όνομα του υποψηφίου. Αν αυτό δεν βρίσκεται στη λίστα των έγκυρων, θα θεωρείται άκυρο και θα αγνοείται. Το πρόγραμμα θα εμφανίζει τα ονόματα που θα αποτελέσουν τα μέλη του 15μελούς συμβουλίου, θεωρώντας ότι δεν υπάρχουν ισοψηφίες και ότι έχουν βάλει υποψηφιότητα περισσότεροι από 15 μαθητές.

82. Σε ένα πρωτάθλημα ποδοσφαίρου συμμετέχουν 10 ομάδες. Η κάθε ομάδα παίζει με κάθε άλλη ομάδα μία φορά. Να αναπτύξετε αλγόριθμο ο οποίος θα διαβάσει τα γκολ που πέτυχε κάθε ομάδα σε κάθε αντίπαλη ομάδα. Αν η νίκη δίνει 3 βαθμούς, η ισοπαλία 1 και η ήττα 0, θα εμφανίζει τα ονόματα των ομάδων από αυτήν με τη μεγαλύτερη βαθμολογία σε αυτήν με τη μικρότερη.

83. Μια καφετέρια έχει 24 τραπέζια των 5 ατόμων. Από το κάθε τραπέζι καθένα άτομο μπορεί να παραγγείλει ένα από τα 15 είδη καφέδων που βρίσκονται στον κατάλογο. Να γράψετε αλγόριθμο ο οποίος θα διαβάσει το όνομα κάθε είδους καφέ καθώς και την τιμή του. Στη συνέχεια θα διαβάσει για κάθε τραπέζι τον αριθμό ατόμων που κάθονται σε αυτό και για καθένα άτομο το όνομα του καφέ που παρήγγειλε. Να εμφανίζει τον αριθμό των τραπεζιών από αυτό με τον μεγαλύτερο λογαριασμό σε αυτό με το μικρότερο.

Παρατήρηση: Θεωρήστε ότι όλες οι τιμές εισόδου είναι έγκυρες.

84. Έστω ότι θέλουμε να διατάξουμε τους μαθητές μιας τάξης κατά φθίνουσα σειρά ύψους. Η τεχνική που θα ακολουθήσουμε είναι η εξής: Αρχικά τοποθετούμε μαθητές σε τυχαία σειρά. Κατόπιν συγκρίνουμε το δεύτερο με το πρώτο και, αν χρειασθεί, τους αντιμεταθέτουμε ώστε ο πρώτος να είναι ο ψηλότερος. Στη συνέχεια θεωρούμε τον τρίτο και τον τοποθετούμε στη σωστή σειρά σε σχέση με τον πρώτο και τον δεύτερο. Κατ' αυτόν τον τρόπο συνεχίζουμε μέχρι να τοποθετήσουμε στη σωστή σειρά όλους τους μαθητές. Να σχεδιάσετε έναν αλγόριθμο που να υλοποιεί αυτή τη μέθοδο ταξινόμησης.