

Δομές δεδομένων



Δομές δεδομένων

Γενικοί ορισμοί

- Με τον όρο δεδομένο δηλώνεται οποιοδήποτε στοιχείο μπορεί να γίνει αντιληπτό, από έναν τουλάχιστον παρατηρητή με μία από τις πέντε του αισθήσεις. Εναλλακτικώς, δεδομένο καλείται η οποιαδήποτε δηλωμένη εκ' των προτέρων κατάσταση στοιχείου, η οποία λαμβάνει χώρα στην τελική πληροφόρηση, μέσω της επεξεργασίας της. Είναι η αφαιρετική αναπαράσταση της πραγματικότητας και συνεπώς μία απλοποιημένη όψη της. Για παράδειγμα, έστω ένα αρχείο μαθητών ενός σχολείου. Τα χρήσιμα δεδομένα που αποθηκεύονται είναι το ονοματεπώνυμο, η ηλικία, το φύλο, η τάξη, το τμήμα κ.λπ., όχι όμως το βάρος, το ύψος κ.λπ. Τα δεδομένα, λοιπόν, είναι ακατέργαστα γεγονότα, και κάθε φορά η επιλογή τους εξαρτάται από τον τύπο του προβλήματος.
- Ο όρος επεξεργασία δεδομένων δηλώνει εκείνη την διαδικασία κατά την οποία ένας 'μηχανισμός' δέχεται δεδομένα, τα επεξεργάζεται με έναν καθορισμένο τρόπο και αποδίδει πληροφορίες. Εναλλακτικώς, η διαδικασία μετασχηματισμού των δεδομένων σε πληροφορία.
- Με τον όρο πληροφορία αναφέρεται οποιοδήποτε γνωσιακό στοιχείο προέρχεται από την επεξεργασία δεδομένων. Η συλλογή των ακατέργαστων δεδομένων και ο συσχετισμός τους δίνει ως αποτέλεσμα την πληροφορία. Με βάση τις δεδομένες πληροφορίες λαμβάνονται διάφορες αποφάσεις και γίνονται ενέργειες. Στη συνέχεια αυτές οι ενέργειες παράγουν νέα δεδομένα, νέες πληροφορίες, νέες αποφάσεις, νέες ενέργειες κ.ο.κ.

Θεωρία Πληροφοριών: Στόχο έχει τη μελέτη της μέτρησης, της κωδικοποίησης και μετάδοσης της πληροφορίας. Η μέτρηση, η κωδικοποίηση, η μετάδοση της πληροφορίας αποτελεί αντικείμενο μελέτης ενός ιδιαίτερου κλάδου, της Θεωρίας Πληροφοριών (Information Theory), που είναι ένα ιδιαίτερα σημαντικό πεδίο της Πληροφορικής.

Σπουδαιότητα των δεδομένων στην Πληροφορική

Όπως η Πληροφορική ορίζεται ως επιστήμη σε συνάρτηση με την έννοια του αλγορίθμου, κατά τον ίδιο τρόπο η Πληροφορική ορίζεται και σε σχέση με την έννοια των δεδομένων. Έτσι, Πληροφορική θεωρείται η επιστήμη που μελετά τα δεδομένα από τις ακόλουθες σκοπιές:

- 1) Υλικού:** Το υλικό (hardware), δηλαδή η μηχανή, επιτρέπει στα δεδομένα ενός προγράμματος να αποθηκεύονται στην κύρια μνήμη και στις περιφερειακές συσκευές του υπολογιστή με διάφορες αναπαραστάσεις (representations). Τέτοιες μορφές είναι η δυαδική, ο κώδικας ASCII (βλ. παράρτημα), ο κώδικας EBCDIC, το συμπλήρωμα του 1 ή του 2 κ.λπ.
- 2) Γλωσσών προγραμματισμού:** Οι διάφορες γλώσσες προγραμματισμού μας επιτρέπουν να χρησιμοποιήσουμε τύπους μεταβλητών για να περιγράψουμε ένα δεδομένο (ακέραιος, πραγματικός, αλφαριθμητικό κλπ). Ο μεταφραστής κάθε γλώσσας φροντίζει για την αποδοτικότερη μορφή αποθήκευσης, από πλευράς υλικού, κάθε μεταβλητής στον υπολογιστή.
- 3) Δομών δεδομένων:** Δομή δεδομένων (data structure) είναι ένα σύνολο δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών επί αυτών. Για παράδειγμα, μία τέτοια δομή είναι η εγγραφή (record), που μπορεί να περιγράψει ένα είδος, ένα πρόσωπο κ.λπ. Η εγγραφή αποτελείται από τα πεδία (fields) που αποθηκεύουν χαρακτηριστικά (attributes) διαφορετικού τύπου, όπως για παράδειγμα ο κωδικός, η περιγραφή κ.λπ. Άλλη μορφή δομής δεδομένων είναι το αρχείο που αποτελείται από ένα σύνολο εγγραφών. Μία επιτρεπτή λειτουργία σε ένα αρχείο είναι η σειριακή προσπέλαση όλων των εγγραφών του.
- 4) Ανάλυσης Δεδομένων:** Τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων μελετώνται έτσι ώστε να αναπαρασταθεί η γνώση για πραγματικά γεγονότα. Οι τεχνολογίες των Βάσεων Δεδομένων (Databases), της Μοντελοποίησης Δεδομένων (Data Modelling) και

της Αναπαράστασης Γνώσης (Knowledge Representation) ανήκουν σε αυτή τη σκοπιά μελέτης των δεδομένων

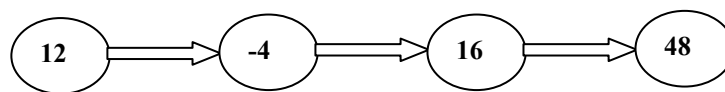
Δομή δεδομένων

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

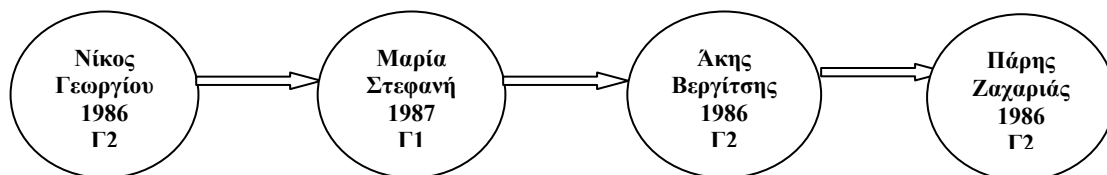
Η δομή αποτελείται από ένα σύνολο κόμβων ή διαφορετικά θέσεων στις οποίες αποθηκεύονται διαφορετικές τιμές. Κάθε κόμβος περιέχει μία απλή τιμή (αριθμό, αλφαριθμητικό κλπ) ή μία πιο σύνθετη τιμή (π.χ. μία εγγραφή).

Ουσιαστικά πρόκειται για μια μεταβλητή η οποία την ίδια χρονική στιγμή μπορεί να έχει πολλές τιμές (διαφορά από προηγούμενες μεταβλητές).

Π.χ. δομή λίστας που περιέχει ακέραιους αριθμούς:



Π.χ. δομή λίστας που περιέχει εγγραφές (Εγγραφή = Μία ομάδα τιμών διαφορετικού τύπου σχετικά με ένα αντικείμενο. Π.χ. μία εγγραφή μαθητή περιέχει το όνομα, επώνυμο, έτος γέννησης, τάξη κλπ).



Βασικές λειτουργίες (πράξεις) επί των δομών:

Προσπέλαση	Είναι η δυνατότητα πρόσβασης σε ένα κόμβο με σκοπό την ανάγνωση ή τροποποίηση του περιεχομένου του.
Αναζήτηση	Προσπελούνται οι κόμβοι της δομής με σκοπό να εντοπιστεί κάποιος που περιέχει μία συγκεκριμένη τιμή ή ιδιότητα.
Εισαγωγή	Προσθήκη νέου κόμβου στη δομή.
Διαγραφή	Αφαίρεση κόμβου από την δομή.
Ταξινόμηση	Διατάσσουμε της κόμβους της δομής σε αύξουσα ή φθίνουσα σειρά (π.χ. αλφαβητικά κατά επώνυμο, όνομα).
Αντιγραφή	Όλοι ή μερικοί κόμβοι αντιγράφονται σε μία άλλη δομή.
Συγχώνευση	Δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
Διαχωρισμός	Μία δομή διασπάται σε δύο ή περισσότερες. Είναι το αντίστροφο της συγχώνευσης.

Σπάνια χρησιμοποιούνται και οι οκτώ λειτουργίες σε κάποια δομή. Συνήθως μια δομή μπορεί να είναι αποδοτικότερη ή λιγότερο αποδοτική από μια άλλη με κριτήριο κάποια λειτουργία.

Για κάθε λειτουργία της δημιουργείται και αλγόριθμος (π.χ. αλγόριθμος αναζήτησης κάποιου στοιχείου). Αρκετές φορές υπάρχουν διαφορετικοί αλγόριθμοι που υλοποιούν μία συγκεκριμένη λειτουργία (π.χ. υπάρχουν αρκετοί αλγόριθμοι για τη λειτουργία της ταξινόμησης). Όταν συμβαίνει αυτό, επιλέγουμε τον αλγόριθμο που είναι πιο αποδοτικός (π.χ. πιο γρήγορος) για τα συγκεκριμένα δεδομένα.

Εξίσωση Wirth (Δημιουργός της γλώσσας Pascal).

Πρόγραμμα = Δομές δεδομένων + Αλγόριθμοι

Σημαίνει ότι για τη δημιουργία ενός προγράμματος πρέπει να ανακαλύψουμε τα δεδομένα, να τα δομήσουμε σε δομές δεδομένων και να τα επεξεργαστούμε με κάποιον αλγόριθμο.

Κατηγορίες δομών δεδομένων

Στατικές	Δυναμικές
Το μέγεθος τους είναι καθορισμένο (σταθερό) και δεν μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος.	Το μέγεθος τους δεν είναι καθορισμένο (δεν είναι σταθερό) και μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. Μπορούν να εισάγονται νέοι κόμβοι και να διαγράφονται υπάρχοντες. (τεχνική της δυναμικής παραχώρησης μνήμης)
Οι κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.	Οι κόμβοι δεν αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.
Είναι πιο εύκολες στην υλοποίησή τους αλλά περιορίζουν τις δυνατότητες ενός προγράμματος γιατί μπορούν να διαχειριστούν σταθερό αριθμό δεδομένων.	Μπορούν να διαχειριστούν μεταβλητό αριθμό δεδομένων αλλά είναι πιο δύσκολες στην διαχείρισή τους στον αλγόριθμο και το πρόγραμμα.

Οι στατικές δομές υλοποιούνται με πίνακες. Ένας πίνακας είναι μια δομή δεδομένων που περιέχει στοιχεία του ίδιου τύπου.

Πίνακες

Πίνακας είναι ένα σύνολο αντικειμένων ίδιου τύπου, τα οποία αναφέρονται με ένα κοινό όνομα. Κάθε ένα από τα αντικείμενα που απαρτίζουν τον πίνακα λέγεται στοιχείο του πίνακα. Η αναφορά σε ατομικά στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από ένα δείκτη. 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).

Στους πίνακες η εισαγωγή και η διαγραφή των κόμβων δεν είναι εφικτή διότι είναι στατικές δομές.

Πλεονεκτήματα / Μειονεκτήματα των πινάκων	
Πλεονεκτήματα	Είναι ένας βολικός κι εύκολος τρόπος διαχείρισης του ίδιου τύπου.
Μειονεκτήματα	1. Απαιτούν μνήμη. Ο πίνακας δεσμεύει από την αρχή του προγράμματος αρκετές θέσεις μνήμης. Έτσι, αν χρησιμοποιούμε πολλούς πίνακες σε ένα πρόγραμμα το επιβαρύνουμε όσον αφορά τη μνήμη. 2. Περιορίζουμε τις δυνατότητες του προγράμματος. Επειδή ο πίνακας είναι στατική δομή, δεν μπορεί να αλλάξει το μέγεθός του κατά την εκτέλεση του προγράμματος.

Πότε είναι προτιμότερο να χρησιμοποιούμε τους πίνακες;

Όταν τα δεδομένα που εισάγουμε, απαιτείται να βρίσκονται στη μνήμη RAM καθ' όλη τη διάρκεια εκτέλεσης του προγράμματος.

Τυπικές επεξεργασίες (Λειτουργίες) σε έναν πίνακα.

1. Διάβασμα των στοιχείων του πίνακα, δηλαδή εισαγωγή τιμών στις θέσεις του.
2. Εκτύπωση των στοιχείων του πίνακα
3. Υπολογισμός του αθροίσματος των στοιχείων του.
4. Υπολογισμός του μέσου όρου των στοιχείων του.
5. Εύρεση του ελάχιστου ή μέγιστου στοιχείου του.
6. Αναζήτηση ενός στοιχείου.
7. Ταξινόμηση του πίνακα.
8. Συγχώνευση δύο πινάκων

Στους αλγόριθμους του πίνακα χρησιμοποιούμε ως επί το πλείστον, τη δομή Για...από...μέχρι

Δομές δεδομένων δευτερεύουσας μνήμης

Σε μεγάλες πρακτικές εμπορικές/επιστημονικές εφαρμογές, το μέγεθος της κύριας μνήμης δεν επαρκεί για την αποθήκευση των δεδομένων. Στην περίπτωση αυτή χρησιμοποιούνται ειδικές δομές για την αποθήκευση των δεδομένων στη δευτερεύουσα μνήμη, δηλαδή κυρίως στο μαγνητικό δίσκο. Οι ειδικές αυτές δομές ονομάζονται **αρχεία** (files). Είναι γνωστό ότι μία σημαντική διαφορά μεταξύ κύριας μνήμης και μαγνητικού δίσκου είναι ότι στην περίπτωση του δίσκου, τα δεδομένα δεν χάνονται, αν διακοπεί η ηλεκτρική παροχή. Έτσι, τα δεδομένα των αρχείων διατηρούνται ακόμη και μετά τον τερματισμό ενός προγράμματος, κάτι που δεν συμβαίνει στην περίπτωση των δομών της κύριας μνήμης, όπως είναι οι πίνακες, όπου τα δεδομένα χάνονται όταν τελειώσει το πρόγραμμα. Τα στοιχεία ενός αρχείου ονομάζονται **εγγραφές** (records), όπου κάθε εγγραφή αποτελείται από ένα ή περισσότερα **πεδία** (fields), που ταυτοποιούν την εγγραφή, και από άλλα πεδία που περιγράφουν διάφορα χαρακτηριστικά της εγγραφής.

Για παράδειγμα, έστω η εγγραφή ενός μαθητή με πεδία: Αριθμός Μητρώου, Ονοματεπώνυμο, Έτος Γέννησης, Τάξη, Τμήμα. Το πεδίο Αριθμός Μητρώου ταυτοποιεί την εγγραφή και ονομάζεται **πρωτεύον κλειδί** (primary key) ή απλά κλειδί. Το πεδίο Ονοματεπώνυμο επίσης ταυτοποιεί την εγγραφή και γι' αυτό αποκαλείται **δευτερεύον κλειδί** (secondary keys), αν υπάρχει πρωτεύον κλειδί. Το πρόβλημα της αναζήτησης (searching) μίας εγγραφής με βάση την τιμή του πρωτεύοντος ή ενός δευτερεύοντος κλειδιού σε αρχεία είναι ιδιαίτερα ενδιαφέρον, αν ληφθεί υπόψη η μεγάλη ποικιλία των χαρακτηριστικών τόσο της δομής (για παράδειγμα, στατική ή δυναμική, τρόπος οργάνωσης, μέσο αποθήκευσης κ.λπ.), του τύπου των δεδομένων (για παράδειγμα, ακέραιοι, κείμενο, χαρτογραφικά δεδομένα, χρονοσειρές κ.λπ.), όσο και της αναζήτησης (δηλαδή, με βάση το πρωτεύον ή το δευτερεύον κλειδί κ.λπ.).

Μονοδιάστατοι πίνακες (μίας γραμμής ή μίας στήλης)

Είναι ουσιαστικά μια μεταβλητή στην οποία κάθε χρονική στιγμή είναι αποθηκευμένες πολλές τιμές. Οι τιμές αυτές είναι διατεταγμένες, δηλαδή υπάρχει μια σειρά, και καθεμία έχει ένα μοναδικό δείκτη που καθορίζει τη θέση του στοιχείου στον πίνακα.

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

Ο πίνακας ορίζεται ως εξής: τύπος όνομα_πίνακα [διαστάσεις].

Π.χ. ακέραιος Π [1:5] δηλώνει έναν μονοδιάστατο πίνακα μίας γραμμής πέντε θέσεων (στήλης) που περιέχει ακεραίους.

Γενικά, ορίζεται ως Π [N].

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

Π.χ. Πίνακας ακεραίων Π [5]

12	-4	22	45	-7
----	----	----	----	----

Π.χ. Πίνακας αλφαριθμητικών Π [4]

"Γιάννης"	"Αλέκα"	"Μάριος"	"Άκης"
-----------	---------	----------	--------

Για να εμφανίσουμε την τιμή που υπάρχει στη θέση 2 αρκεί να εκτελέσουμε την εντολή Εμφάνισε Π[2], ενώ για να εκχωρήσουμε την τιμή 15 στην ίδια θέση πρέπει να εκτελέσουμε την εντολή Π[2]← 15.

Παρατηρήσεις

1. Το όνομα ενός πίνακα ακολουθεί τους ίδιους κανόνες ονοματολογία όπως και οι απλές μεταβλητές. Αποδεκτά ονόματα είναι: Α, Β, Μισθός, Άθροισμα, κ.α.
2. Το πλήθος των στοιχείων ενός πίνακα είναι σταθερό και καθορισμένο από την αρχή του αλγορίθμου διότι είναι στατική δομή.
3. Στον πίνακα αναφερόμαστε με το όνομά του (πχ πίνακας Π), ενώ σε ένα στοιχείο του με το όνομα του πίνακα και το δείκτη του, δηλαδή τη θέση του (πχ Π[1]).
4. Για να επεξεργαστούμε ένα πίνακα σε έναν αλγόριθμο, πρέπει να επεξεργαστούμε κάθε στοιχείο του χωριστά. Δηλαδή για να εμφανίσουμε τις τιμές ενός πίνακα 10 θέσεων χρειαζόμαστε 10 εντολές Εμφάνισε ή μια επανάληψη 10 φορές.
5. κάθε πίνακας επιτρέπεται να περιέχει στοιχεία μόνο του ίδιου τύπου και συνήθως αναφέρονται μαζί με το όνομά του οι θέσεις του πίνακα και ο τύπος του, πχ πίνακας Π ακεραίων αριθμών 10 θέσεων.
6. Οι τιμές δύο ή περισσότερων στοιχείων ενός πίνακα μπορεί να είναι και ίδιες.
7. Οι πίνακες μίας διάστασης χρησιμοποιούν έναν δείκτη για τον προσδιορισμό της θέσης, πχ Π[1]. Το 1 είναι ο δείκτης θέσης.
8. Ο δείκτης είναι μια μεταβλητή που μπορεί να έχει οποιοδήποτε δεκτό όνομα. Συνήθως χρησιμοποιούνται ως δείκτες οι μεταβλητές: i, j, k.

Όταν θέλουμε να επεξεργαστούμε μαζικά τους πίνακες χρησιμοποιούμε τη δομή επανάληψης **Για..από.. μέχρι**

```
Πχ. ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
      ΔΙΑΒΑΣΕ Π[Ι]
      ΓΡΑΨΕ Π[Ι]
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Για το διάβασμα και την εμφάνιση των στοιχείων του πίνακα μπορούμε να χρησιμοποιήσουμε αντίστοιχα τα Δεδομένα // Π// και τα Αποτελέσματα //Π// στην αρχή και στο τέλος του αλγορίθμου αντί για την επανάληψη Για...

Λυμένα Παραδείγματα

1. Έστω ένας πίνακας Π 100 ακεραίων και Ον πίνακας 100 ονομάτων. Πώς δηλώνονται;

ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: Π [100]
ΧΑΡΑΚΤΗΡΕΣ: ΟΝ [100]

2. Δίνεται ο παρακάτω πίνακας

8	5	1	-2	30	-7
---	---	---	----	----	----

Να γράψετε ποιο θα είναι το αποτέλεσμα μετά την εκτέλεση των εντολών

α. ΓΡΑΨΕ Α[3]
β. $I \leftarrow 4$ ΓΡΑΨΕ Α[I]
γ. $I \leftarrow 2$ ΓΡΑΨΕ Α[I+3]
δ. $A[1] \leftarrow 2$
ε. $A[2] \leftarrow A[6] + A[5]$
στ. $A[4] \leftarrow 2$ ΓΡΑΨΕ Α[4]

α. ΘΑ ΕΜΦΑΝΙΣΤΕΙ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΤΡΙΤΟΥ ΣΤΟΙΧΕΙΟΥ ΤΟΥ ΠΙΝΑΚΑ Α, ΔΗΛΑΔΗ Ο ΑΡΙΘΜΟΣ 1.

β. ΤΟ I ΛΑΜΒΑΝΕΙ ΤΗΝ ΤΙΜΗ 4 ΑΡΑ ΕΜΦΑΝΙΣΤΕΙ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΤΕΤΑΡΤΟΥ ΣΤΟΙΧΕΙΟΥ ΤΟΥ ΠΙΝΑΚΑ Α, ΔΗΛΑΔΗ Ο ΑΡΙΘΜΟΣ -2.

γ. ΤΟ I ΛΑΜΒΑΝΕΙ ΤΟΝ ΑΡΙΘΜΟ 2 ΚΑΙ ΣΤΗ ΣΥΝΕΧΕΙΑ, ΘΑ ΕΜΦΑΝΙΣΤΕΙ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΣΤΟΙΧΕΙΟΥ Α[2+3] ΔΗΛΑΔΗ ΤΟ Α[5], ΠΟΥ ΕΙΝΑΙ Ο ΑΡΙΘΜΟΣ 30.

δ. Ο ΑΡΙΘΜΟΣ 2 ΤΟΠΟΘΕΤΕΙΤΑΙ ΣΤΟ 1^ο ΣΤΟΙΧΕΙΟ ΤΟΥ ΠΙΝΑΚΑ ΠΟΥ ΘΑ ΑΝΤΙΚΑΤΑΣΤΗΣΕΙ ΤΟΝ ΑΡΙΘΜΟ 8 Ο ΟΠΟΙΟΣ ΥΠΗΡΧΕ ΜΕΧΡΙ ΤΩΡΑ.

ε. ΥΠΟΛΟΓΙΖΕΤΑΙ Η ΕΚΦΡΑΣΗ ΠΟΥ ΥΠΑΡΧΕΙ ΔΕΞΙΑ ΑΠΟ ΤΗΝ ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ ΚΑΙ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΤΗΣ ΠΡΟΣΘΕΣΗΣ ΤΟΠΟΘΕΤΕΙΤΑΙ ΣΤΟ 2^ο ΣΤΟΙΧΕΙΟ ΤΟΥ ΠΙΝΑΚΑ, ΔΗΛΑΔΗ $A[2] \leftarrow A[6] + A[5]$, $A[2] \leftarrow -7 + 30$, $A[2] \leftarrow 23$.

στ. Ο ΑΡΙΘΜΟΣ 2 ΚΑΤΑΧΩΡΙΖΕΤΑΙ ΣΤΟ 4^ο ΣΤΟΙΧΕΙΟ ΚΑΙ Η ΕΠΟΜΕΝΗ ΕΝΤΟΛΗ ΕΜΦΑΝΙΖΕΙ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ 4^{ου} ΣΤΟΙΧΕΙΟΥ ΔΗΛΑΔΗ ΤΟΝ ΑΡΙΘΜΟ 2.

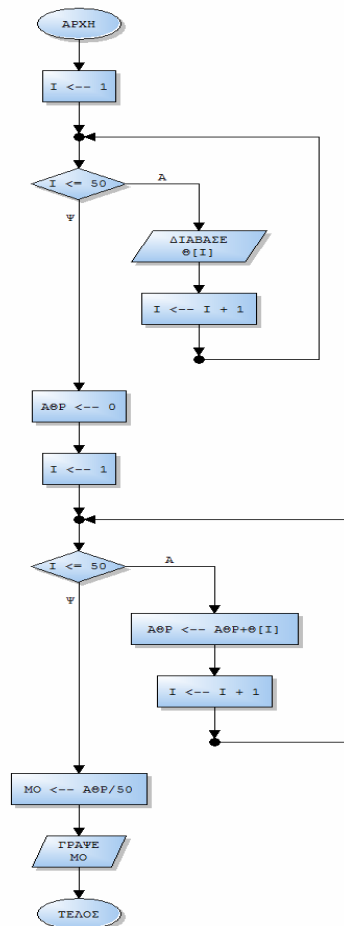
3. Να δημιουργηθεί με αλγόριθμο ο ακόλουθος πίνακας 1000 θέσεων

1	2	3	...	1000
---	---	---	-----	------

ΑΛΓΟΡΙΘΜΟΣ ΑΣΚ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 1000
 Π[I] ← I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ ΑΣΚ

4. Να μετατρέψετε το παρακάτω πρόγραμμα στο ισοδύναμο διάγραμμα ροής

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
 ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: Θ[50], ΑΘΡ, Ι
 ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
 ΑΡΧΗ
 ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
 ΔΙΑΒΑΣΕ Θ[Ι]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΑΘΡ←0
 ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
 ΑΘΡ←ΑΘΡ+Θ[Ι]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΜΟ←ΑΘΡ/50
 ΓΡΑΨΕ ΜΟ
 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ



5. Δίνεται ο πίνακας Β. Τι θα εμφανίσει το παρακάτω τμήμα εντολών

-4	6	-1	2	8	3	1	5
----	---	----	---	---	---	---	---

ΓΡΑΨΕ Β[1]
 Α←Β[1]
 ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 8
 ΑΝ Ι MOD 2 =1 ΤΟΤΕ
 Α←Α+Β[Ι]
 ΑΛΛΙΩΣ
 ΓΡΑΨΕ Β[Ι]+Ι
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ Α

Ι	Α	ΟΘΟΝΗ
		-4
	-4	
2		8
3	-5	
4		6
5	3	
6		9
7	4	
8		13
		4

6. Να συμπληρώσετε τα κενά στο πρόγραμμα ώστε να διαβάσει 20 ακέραιους, να τους καταχωρίζει στον πίνακα A και να εμφανίζει τον μέσο όρο του πίνακα

```
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    ΔΙΑΒΑΣΕ A [(1)]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Σ ← (2)
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    Σ ← Σ + (3) [(4)]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Μ ← (5) / 20
ΓΡΑΨΕ (6)
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

1 -----I
2-----0
3-----A
4-----I
5-----Σ
6-----M

7. Να γραφεί πρόγραμμα που θα διαβάσει έναν πίνακα 50 αριθμών και στη συνέχεια υπολογίζει και εμφανίζει: α) το άθροισμα των στοιχείων του, β) το πλήθος των μη μηδενικών του στοιχείων και γ) το ποσοστό των μηδενικών στοιχείων.

```
ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, ΠΛ
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[50], ΑΘΡ, ΜΔΣ, ΠΟΣ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
    ΔΙΑΒΑΣΕ Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΘΡ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
    ΑΘΡ ← ΑΘΡ + Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΑΘΡ
ΠΛ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
    ΑΝ Π[I] <> 0 ΤΟΤΕ
        ΠΛ ← ΠΛ + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ
ΜΔΣ ← 50 - ΠΛ
ΠΟΣ ← ΜΔΣ * 100 / 50
ΓΡΑΨΕ ΠΟΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ
```

8. Θεωρώντας δεδομένο πίνακα N αριθμών να δημιουργηθεί τμήμα προγράμματος που θα ελέγχει αν όλοι οι αριθμοί του είναι ακέραιοι ή όχι. Να εμφανίζει «Όλοι ακέραιοι» ή «Δεν είναι όλοι ακέραιοι» ανάλογα με το τι ισχύει.

1Η ΛΥΣΗ:

```
ΠΛ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΑΝ Π [Ι] = Α_Μ(Π[Ι]) ΤΟΤΕ
        ΠΛ ← ΠΛ + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΠΛ = Ν ΤΟΤΕ
    ΓΡΑΨΕ 'ΟΛΟΙ ΑΚΕΡΑΙΟΙ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΔΕΝ ΕΙΝΑΙ ΟΛΟΙ ΑΚΕΡΑΙΟΙ'
ΤΕΛΟΣ_ΑΝ
```

2Η ΛΥΣΗ:

```
Χ ← ΑΛΗΘΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΑΝ Π[Ι] <> Α_Μ(Π[Ι]) ΤΟΤΕ
        Χ ← ΨΕΥΔΗΣ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ Χ = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'ΟΛΟΙ ΑΚΕΡΑΙΟΙ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΔΕΝ ΕΙΝΑΙ ΟΛΟΙ ΑΚΕΡΑΙΟΙ'
ΤΕΛΟΣ_ΑΝ
```

9. Να γραφεί πρόγραμμα που θα διαβάζει τις ηλικίες 30 μαθητών σε πίνακα. Να εμφανιστεί η μικρότερη και η μεγαλύτερη ηλικία. Επίσης να υπολογιστεί και να εμφανιστεί ο αριθμός των μαθητών που έχουν την μικρότερη ηλικία.

```
ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, ΠΛ
    ΠΡΑΓΜΑΤΙΚΕΣ: Η[30], ΜΙΝ, ΜΑΧ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΔΙΑΒΑΣΕ Η[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΙΝ ← Η[1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΑΝ Η[Ι] < ΜΙΝ ΤΟΤΕ
        ΜΙΝ ← Η[Ι]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΙΝ
ΜΑΧ ← Η[1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΑΝ Η[Ι] > ΜΑΧ ΤΟΤΕ
        ΜΑΧ ← Η[Ι]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ
```

```

ΠΛ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΑΝ Η[Ι] = ΜΙΝ ΤΟΤΕ
        ΠΛ ← ΠΛ + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

10. Σε τρεις πίνακες διαβάζουμε το επίθετο, την ηλικία και το φύλο 80 μαθητών. Το φύλο μπορεί να είναι Α ή Κ και η ηλικία αριθμός από το 10 έως το 18 (τα άκρα δεν περιλαμβάνονται). Να γραφεί πρόγραμμα που θα: α) διαβάσει τους πίνακες. Κατά την εισαγωγή των στοιχείων να γίνεται έλεγχος ορθής καταχώρησης φύλου και ηλικίας. β) υπολογίζει το ποσοστό των αγοριών και το ποσοστό των κοριτσιών και τα εμφανίζει μετά τις λέξεις ΑΓΟΡΙΑ και ΚΟΡΙΤΣΙΑ αντίστοιχα. γ) Εμφανίζει το επώνυμο του πιο νέου μαθητή.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, ΠΑ, ΠΚ
    ΠΡΑΓΜΑΤΙΚΕΣ: Η[80], ΠΟΣΑ, ΠΟΣΚ, ΜΙΝ
    ΧΑΡΑΚΤΗΡΕΣ: Ε[80], Φ[80], ΕΜΙΝ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 80
    ΔΙΑΒΑΣΕ Ε[Ι]
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ Η[Ι]
        ΜΕΧΡΙΣ_ΟΤΟΥ Η[Ι] > 10 ΚΑΙ Η[Ι] < 18
        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            ΔΙΑΒΑΣΕ Φ[Ι]
            ΜΕΧΡΙΣ_ΟΤΟΥ Φ[Ι] = 'Α' Η' Φ[Ι] = 'Κ'
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΠΑ ← 0
    ΠΚ ← 0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 80
        ΑΝ Φ[Ι] = 'Α' ΤΟΤΕ
            ΠΑ ← ΠΑ + 1
        ΑΛΛΙΩΣ
            ΠΚ ← ΠΚ + 1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΠΟΣΑ ← ΠΑ * 100 / 80
    ΠΟΣΚ ← ΠΚ * 100 / 80
    ΓΡΑΨΕ 'ΑΓΟΡΙΑ', ΠΟΣΑ
    ΓΡΑΨΕ 'ΚΟΡΙΤΣΙΑ', ΠΟΣΚ
    ΜΙΝ ← Η[1]
    ΕΜΙΝ ← Ε[1]
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 80
        ΑΝ Η [Ι] < ΜΙΝ ΤΟΤΕ
            ΜΙΝ ← Η[Ι]
            ΕΜΙΝ ← Ε[Ι]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ ΕΜΙΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

ΕΝΑΛΛΑΚΤΙΚΟΣ ΤΡΟΠΟΣ ΥΠΟΛΟΓΙΣΜΟΥ ΤΟΥ ΕΡΩΤΗΜΑΤΟΣ Γ
ΕΜΦΑΝΙΣΗ ΟΛΩΝ ΤΩΝ ΘΕΣΕΩΝ ΠΟΥ ΥΠΑΡΧΕΙ ΤΟ ΜΙΝ

```

MIN ← 100
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 80
    ΑΝ Η[I] < MIN ΤΟΤΕ
        MIN ← Η[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 80
    ΑΝ Η[I] = MIN ΤΟΤΕ
        ΓΡΑΨΕ Ε[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

11. Μια εταιρεία πώλησης αυτοκινήτων καταχωρεί σε πίνακα 35 θέσεων τις τιμές των μοντέλων αυτοκινήτων που διακινεί και σε ένα δεύτερο πίνακα τις ονομασίες τους. Να γραφεί πρόγραμμα που: α) διαβάζει τους πίνακες, β) δημιουργεί νέο πίνακα που θα περιέχει σε κάθε θέση του μια από τις λέξεις «φθηνό», «κανονικό», «ακριβό» ανάλογα με το αν η τιμή του είναι μικρότερη από 15000€, μεταξύ 15000 και 25000€, μεγαλύτερη ή ίση 25001€ αντίστοιχα. γ) υπολογίζει και εμφανίζει το ποσοστό των ακριβών αυτοκινήτων, δ) υπολογίζει και εμφανίζει το μέσο όρο πώλησης των φθηνών αυτοκινήτων και ε) υπολογίζει και εμφανίζει τη μέγιστη τιμή πώλησης των αυτοκινήτων με κανονική τιμή και την ονομασία του.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, ΠΛ, ΠΛ2
    ΠΡΑΓΜΑΤΙΚΕΣ: Τ[35], ΠΟΣ, ΑΘΡ, ΜΑΧ, ΜΟ
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[35], Μ[35], ΟΜΑΧ
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 35
    ΔΙΑΒΑΣΕ Τ[I], Ο[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 35
    ΑΝ Τ[I]<15000 ΤΟΤΕ
        Μ[I] ← 'ΦΘΗΝΟ'
    ΑΛΛΙΩΣ_ΑΝ Τ[I]<=25000 ΤΟΤΕ
        Μ[I] ← 'ΚΑΝΟΝΙΚΟ'
    ΑΛΛΙΩΣ
        Μ[I] ← 'ΑΚΡΙΒΟ'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛ ← 0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 35
    ΑΝ Μ[I]='ΑΚΡΙΒΟ' ΤΟΤΕ
        ΠΛ ← ΠΛ + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΟΣ ← ΠΛ*100/35
ΓΡΑΨΕ ΠΟΣ
ΑΘΡ ← 0
ΠΛ2 ← 0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 35
    ΑΝ Μ[I]='ΦΘΗΝΟ' ΤΟΤΕ
        ΑΘΡ ← ΑΘΡ + Τ[I]
        ΠΛ2 ← ΠΛ2 + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΑΝ ΠΛ2<>0 ΤΟΤΕ
    ΜΟ ← ΑΘΡ/ΠΛ2
    ΓΡΑΨΕ ΜΟ
ΤΕΛΟΣ_ΑΝ
ΜΑΧ ← -1
ΟΜΑΧ ← ''
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 35
    ΑΝ Μ[Ι]= 'ΚΑΝΟΝΙΚΟ' ΚΑΙ Τ[Ι] > ΜΑΧ ΤΟΤΕ
        ΜΑΧ ← Τ[Ι]
        ΟΜΑΧ ← Ο[Ι]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ, ΟΜΑΧ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

12. Σε ένα βιβλιοπωλείο υπάρχουν 200 διαφορετικά βιβλία με θέμα την πληροφορική. Να γραφεί πρόγραμμα που: α. Για το κάθε βιβλίο 1. διαβάσει το τίτλο του και θα τον καταχωρίσει στο πίνακα ΤΙΤΛΟΙ, 2. θα διαβάσει την τιμή πώλησης και θα την καταχωρίσει στον πίνακα ΤΙΜΗ, εξασφαλίζοντας ότι η τιμή είναι θετικός αριθμός. β. Θα υπολογίζει και εμφανίζει 1. το πλήθος των βιβλίων με τιμή πώλησης τουλάχιστον 40€, 2. το συνολικό ποσό που θα εισπράξει το βιβλιοπωλείο από την πώληση όλων των βιβλίων. γ. Θα εμφανίζει 1. τους τίτλους των βιβλίων με τιμή από 17 έως 20 €, 2. τους τίτλους των βιβλίων με τιμή πώλησης μικρότερη από τη μέση τιμή πώλησης όλων των βιβλίων.

```

ΠΡΟΓΡΑΜΜΑ ΒΙΒΛΙΟΠΩΛΕΙΟ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, ΠΛΗΘΟΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΤΙΜΗ[200], ΜΤ, ΣΥΝΟΛΟ,
    ΧΑΡΑΚΤΗΡΕΣ: ΤΙΤΛΟΙ[200]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 200
    ΔΙΑΒΑΣΕ ΤΙΤΛΟΙ[Ι]
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ ΤΙΜΗ[Ι]
        ΜΕΧΡΙΣ_ΟΤΟΥ ΤΙΜΗ[Ι]>0
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΠΛΗΘΟΣ← 0
    ΣΥΝΟΛΟ← 0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 200
        ΣΥΝΟΛΟ ← ΣΥΝΟΛΟ + ΤΙΜΗ[Ι]
        ΑΝ ΤΙΜΗ[Ι] >40 ΤΟΤΕ
            ΠΛΗΘΟΣ← ΠΛΗΘΟΣ + 1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ ΠΛΗΘΟΣ, ΣΥΝΟΛΟ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 200
        ΑΝ ΤΙΜΗ[Ι] >=17 ΚΑΙ ΤΙΜΗ[Ι] <=20 ΤΟΤΕ
            ΓΡΑΨΕ ΤΙΤΛΟΙ[Ι]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΤ← ΣΥΝΟΛΟ /200
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 200
        ΑΝ ΤΙΜΗ[Ι] < ΜΤ ΤΟΤΕ
            ΓΡΑΨΕ ΤΙΤΛΟΙ[Ι]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΒΙΒΛΙΟΠΩΛΕΙΟ

```

13. Μια εταιρεία έχει δύο καταστήματα με 40 και 35 υπαλλήλους αντίστοιχα. Η μηχανοργάνωση έχει καταχωρίσει στον πίνακα M1 τους μισθούς των υπαλλήλων του 1^{ου} καταστήματος και στον M2 του 2^{ου} καταστήματος. Να γραφεί πρόγραμμα που με δεδομένους τους παραπάνω πίνακες να βρίσκει πόσοι υπάλληλοι αμείβονται με τον μεγαλύτερο μισθό και από τα δύο καταστήματα.

```

ΠΡΟΓΡΑΜΜΑ ΕΤΑΙΡΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, ΠΛΗΘ, Κ
    ΠΡΑΓΜΑΤΙΚΕΣ: M1[40], M2[35], ΝΠ[75], ΜΑΧ
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 40
    ΔΙΑΒΑΣΕ M1[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 35
    ΔΙΑΒΑΣΕ M2[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 40
    ΝΠ[I]←M1[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Κ←0
ΓΙΑ I ΑΠΟ 41 ΜΕΧΡΙ 75
    Κ←Κ+1
    ΝΠ[I]←M2[Κ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ←ΝΠ[1]
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 75
    ΑΝ ΝΠ[I]> ΜΑΧ ΤΟΤΕ
        ΜΑΧ←ΝΠ[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛΗΘ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 75
    ΑΝ ΝΠ[I]=ΜΑΧ ΤΟΤΕ
        ΠΛΗΘ←ΠΛΗΘ+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛΗΘ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΕΤΑΙΡΙΑ

```

Ασκήσεις

1. Δίνεται το παρακάτω τμήμα εντολών, όπου κάθε εντολή περιέχει κενά που αντιστοιχούν σε σταθερά, μεταβλητή, τελεστή. Επίσης ο πίνακας που απεικονίζει την μνήμη όπου κάθε γραμμή του αντιστοιχεί στην διπλανή εντολή του τμήματος και η κάθε στήλη σε μεταβλητή. Η κάθε γραμμή δείχνει ποια μεταβλητή λαμβάνει τιμή από την εκτέλεση της διπλανής εντολής.

1. A←.....
2. B←8+.....
3. I←.....
4. Π[.....]← -
5. I←I1

- 6. $\Pi[\dots] \leftarrow \dots$
- 7. $\Pi[I+\dots] \leftarrow \Pi[\dots]+10$
- 8. $I \leftarrow I \dots 1$
- 9. $\Pi[I+\dots] \leftarrow 2 * \Pi[I-\dots] - \Pi[I-\dots]$

	A	B	I	$\Pi[1]$	$\Pi[2]$	$\Pi[3]$	$\Pi[4]$
1	7						
2		15					
3			1				
4				8			
5			2				
6					15		
7						25	
8			3				
9							1

2. Δίνεται ο πίνακας A

-5	7	2	9	-10	3	6
----	---	---	---	-----	---	---

Ποιο θα είναι το περιεχόμενο του πίνακα A, μετά την εκτέλεση του παρακάτω προγράμματος

ΑΡΧΗ

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 7

 ΑΝ $A[I] > 5$ ΚΑΙ $A[I] < 8$ ΤΟΤΕ

$A[I] \leftarrow 1 + A[I] \text{ DIV } 3$

 ΑΛΛΙΩΣ_ΑΝ $A[I] > 0$ ΚΑΙ $A[I] \leq 5$ ΤΟΤΕ

$A[I] \leftarrow A[I+1] \text{ MOD } A[I-1]$

 ΑΛΛΙΩΣ

$A[I] \leftarrow (-1 * A[I+2]) - A[I]$

 ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

3. Ποια θα είναι τα περιεχόμενα του πίνακα A μετά την εκτέλεση του παρακάτω τμήματος;

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5

$A[I] \leftarrow I$

```

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 5
  ΑΝ (I MOD 2 = 0) ΤΟΤΕ
    A[I] ← 2 * A[I - 1] + 1
  ΑΛΛΙΩΣ
    A[I] ← A[I] + A[I - 1]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

4. Να γραφεί τμήμα προγράμματος που θα δημιουργεί τους παρακάτω πίνακες

2	4	6	8	10	12	14	16	18	20
---	---	---	---	----	----	----	----	----	----

512	256	128	64	32	16	8	4	2	1
-----	-----	-----	----	----	----	---	---	---	---

0	1	0	1	0	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---

11	13	15	17	19	21	10	12	14	16	18	20
----	----	----	----	----	----	----	----	----	----	----	----

5. Να γίνει το διάγραμμα ροής του τμήματος

```

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100
  ΔΙΑΒΑΣΕ A[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100
  ΑΝ A[I]>0 ΤΟΤΕ
    ΠΛ←ΠΛ+1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ

```

6. Να αναπτύξετε πρόγραμμα που θα διαβάζει έναν μονοδιάστατο πίνακα αριθμών θα υπολογίζει και θα εμφανίζει τον μικρότερο αριθμό καθώς και τη θέση του στον πίνακα.

7. Ο καθηγητής πληροφορικής θέλει να επεξεργαστεί στατιστικά την απόδοση των 1500 μαθητών στο μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον. Από τη μηχανογράφηση του σχολείου λαμβάνονται με ηλεκτρονικό τρόπο οι προφορικοί βαθμοί των δυο τετράμηνων και οι γραπτοί βαθμοί μαθητών στις εξετάσεις. Έχοντας υπόψη ότι τα ποσοστά συμμετοχής των παραπάνω στο βαθμό πρόσβασης είναι 30% και 70 % αντίστοιχα, να αναπτυχθεί πρόγραμμα που: Θα διαβάζει τα ονόματα και τους βαθμούς των μαθητών και θα καταχωρίζει στους αντίστοιχους πίνακες. Θα υπολογίζει και θα εκτυπώνει τους βαθμούς πρόσβασης όλων των μαθητών. Θα εκτυπώνει τα ονόματα των μαθητών με βαθμό πρόσβασης μικρότερο από 9.5. Θα εκτυπώνει τα ονόματα των μαθητών με βαθμό πρόσβασης μεγαλύτερο από 18. Ποιός είναι ο μέγιστος βαθμός πρόσβασης; Πόσοι μαθητές έχουν βαθμός πρόσβασης ίσο με τον μέγιστο;

8. Ένας κτηνοτρόφος χρησιμοποιεί τρεις διαφορετικούς τύπους τροφής, 'Α', 'Β' και 'Γ' για το τάισμα των ζώων του και την πρώτη μέρα κάθε μήνα αγοράζει τις κατάλληλες ποσότητες από τον κάθε τύπο τροφής που θα του χρειαστούν. Να γραφεί πρόγραμμα που: α. θα διαβάζει την τιμή ανά κιλό που αγοράζει για τον κάθε τύπο τροφής, β. για κάθε μήνα του περασμένου έτους, να διαβάζει τις ποσότητες σε κιλά που αγόρασε από τον κάθε τύπο τροφής και να τις τοποθετεί σε μονοδιάστατους πίνακες, γ. να δημιουργεί τον πίνακα ΣΠ, που κάθε στοιχείο θα περιέχει τα χρήματα που ξόδεψε ο κτηνοτρόφος για την αγορά όλων των τύπων τροφών, για ένα μήνα και να εμφανίζει δ. το μέσο μηνιαίο ποσό χρημάτων που ξόδεψε ο κτηνοτρόφος για την αγορά και των τριών ειδών, τους μήνες (αύξων αριθμός μήνα) που ξόδεψε περισσότερα χρήματα από το μέσο μηνιαίο ποσό χρημάτων, τον τύπο της τροφής από τον οποίο αγόρασε τα περισσότερα κιλά στο διάστημα του περασμένου έτους

9. Μια τράπεζα διαθέτει πελατολόγιο 15000 κατόχων πιστωτικής κάρτας σε ολόκληρη την Ελλάδα. Να αναπτύξετε πρόγραμμα που με δεδομένα τα στοιχεία των πελατών της τράπεζας και των οφειλών τους θα εκτυπώνει:

- i. Τα ονόματα των πελατών της τράπεζας με οφειλές πάνω του μέσου όρου
- ii. Τα ονόματα των πελατών με μηδενικές οφειλές
- iii. Ποιά είναι η μεγαλύτερη οφειλή προς την τράπεζα
- iv. Ποιοι πελάτες έχουν οφειλή ίση με την μέγιστη

10. Το τμήμα μισθοδοσίας καταχωρεί τις εισπράξεις μιας αλυσίδας των 30 καταστημάτων που διαθέτει σε έναν πίνακα. Αντίστοιχα, σε έναν πίνακα 30 θέσεων καταχωρούνται τα ονόματα - επωνυμία των καταστημάτων. Να αναπτυχθεί πρόγραμμα όπου:

- i. Να εκτυπώνει το όνομα του καταστήματος με τις μεγαλύτερες εισπράξεις
- ii. Να εκτυπώνει το όνομα του καταστήματος με τις μικρότερες εισπράξεις
- iii. Να υπολογίζει και να εκτυπώνει το σύνολο των εισπράξεων της εταιρείας και τον μέσο όρο για κάθε κατάσταση

11. Σε ένα θερμοκήπιο η ημερήσια ποσότητα νερού που απαιτείται καταγράφεται σε ηλεκτρονικό αρχείο. Να γραφεί πρόγραμμα που: θα διαβάσει τον αριθμό των ημερών κάθε μήνα του προηγούμενου έτους και θα τον καταχωρίζει στον πίνακα M[12]. Για κάθε μέρα του έτους θα ζητάει την ποσότητα νερού σε κιλά και θα την καταχωρίζει στον πίνακα N. Θα υπολογίζει και εμφανίζει: την συνολική ποσότητα νερού σε κιλά για όλο το έτος, την συνολική ποσότητα νερού σε κιλά για το πρώτο εξάμηνο του έτους, το τρίμηνο του έτους που σημειώθηκε η μεγαλύτερη κατανάλωση. Να εμφανίζει τον αριθμό της ημέρας του έτους κατά την οποία η κατανάλωση του νερού ήταν μεγαλύτερη από την κατανάλωση της προηγούμενης και της επόμενης ημέρας.

12. Σε ένα νοσοκομείο υπάρχουν 10 κλινικές οι οποίες είναι αριθμημένες από το 1 έως το 10 και οι ονομασίες είναι στο πίνακα ΚΛΙΝΙΚΕΣ. Κατά το πρώτο τρίμηνο του έτους εισήχθησαν N ασθενείς, με τον καθένα να νοσηλεύτηκε σε μια κλινική. Να γραφεί πρόγραμμα που: Θα διαβάσει τον αριθμό N των ασθενών, εξασφαλίζοντας ότι είναι μεγαλύτερος του 10 και μικρότερος του 100. Για κάθε ασθενή να διαβάσει τον αριθμό της κλινικής που νοσηλεύτηκε και να τον αποθηκεύει στον πίνακα AP. Θα διαβάσει τις ονομασίες των 10 κλινικών και θα τις αποθηκεύει στον πίνακα ΚΛΙΝΙΚΕΣ. Να εμφανίζει το όνομα της κάθε κλινικής και τον αριθμό των ασθενών που νοσηλεύτηκαν σε αυτή το πρώτο τρίμηνο. Να εμφανίζει την ονομασία της κλινικής στην οποία νοσηλεύτηκαν οι περισσότεροι ασθενείς στο συγκεκριμένο διάστημα.

Αναζήτηση στοιχείου

Η λειτουργία της αναζήτησης είναι η εύρεση της θέσης στην οποία υπάρχει μια συγκεκριμένη τιμή. Οι πιο γνωστές μέθοδοι είναι

- η σειριακή (γραμμική) και
- η δυαδική.

Σειριακή μέθοδος

Είναι αρκετά απλή στην υλοποίηση αλλά όχι τόσο αποδοτική (γρήγορη).

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.

Πότε χρησιμοποιούμε τη μέθοδο αυτή;

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

Θα χρειαστούμε μία **μεταβλητή βρέθηκε** λογικού τύπου, που θα γίνει True αν το στοιχείο βρεθεί. Επίσης, δεν γνωρίζουμε σε ποια θέση βρίσκεται. Μπορεί να είναι στην 1^η θέση αλλά μπορεί να είναι στη Νστή θέση. Θα χρησιμοποιήσουμε και μία **μεταβλητή θέση** που θα κρατήσει τη θέση όπου βρέθηκε.

Τη χρησιμοποιούμε όταν θέλουμε να βρούμε αν υπάρχει ή όχι ένα συγκεκριμένο στοιχείο μέσα σε έναν πίνακα. Κατά τη σειριακή αναζήτηση, ένα – ένα τα στοιχεία του πίνακα ελέγχονται αν είναι αυτό που ψάχνουμε. Η αναζήτηση συνεχίζεται μέχρι να βρεθεί το στοιχείο ή μέχρι να τελειώσει ο πίνακας.

Ο αλγόριθμος της σειριακής αναζήτησης έχει ως εξής:

Αλγόριθμος **Σειριακή Αναζήτηση**

Δεδομένα // N , table, key //

βρέθηκε \leftarrow Ψευδής

θέση \leftarrow 0

$i \leftarrow 1$

Όσο ($i \leq N$) και (βρέθηκε = Ψευδής) επανάλαβε

 Αν $Π[i] = key$ τότε

 βρέθηκε \leftarrow Αληθής

 θέση $\leftarrow i$

 αλλιώς

$i \leftarrow i + 1$

 τέλος_αν

τέλος_επανάληψης

Αν βρέθηκε = Αληθής τότε

 Εμφάνισε 'Το στοιχείο βρέθηκε στη θέση', θέση

αλλιώς

 Εμφάνισε 'Το στοιχείο δεν βρέθηκε'

τέλος_αν

Τέλος Αναζήτηση

!N: διάσταση του πίνακα, table: πίνακας, key: τιμή αναζήτησης

!αρχικοποίηση μεταβλητών

!όσο δεν είμαστε στο τέλος του πίνακα και δεν βρέθηκε

!αν το I στοιχείο ισούται με αυτό που ψάχνουμε τότε

!βρέθηκε

!βάλε τον αριθμό θέσης στην μεταβλητή θέση

!αλλιώς

!προχώρα στην επόμενη θέση

Ή εναλλακτικά χρησιμοποιώντας τη δομή Μέχρις_Ότου Αρχή_επανάληψης

Αν (table[i] = key) τότε

done ← αληθής

position ← i

Αλλιώς

i ← i + 1

Τέλος_αν

Μέχρις_Ότου (done = αληθής) ή (i > n)

Διαδική μέθοδος

Η διαδική είναι πιο αποδοτική αλλά απαιτείται ο πίνακας να είναι ταξινομημένος.

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

Δεδομένα // N, A, S //

Left ← 1 *! αριστερό όριο (αρχικός)*

Right ← N *! δεξιό όριο (τελικός)*

pos ← 0 *! θέση του στοιχείου*

Flag ← ψευδής *! θεωρούμε ότι δεν υπάρχει στον πίνακα*

Όσο (Left ≤ Right) και (Flag = ψευδής) επανάλαβε

M ← (Left+Right) div 2 *! δείκτης για το μέσο του διαστήματος*

Αν A[M]=Key τότε *! το μεσαίο στοιχείο ίσο με το ζητούμενο*

pos ← M *! θέση στο μέσο*

Flag ← αληθής

Αλλιώς_Αν A[M]< Key τότε *! αν το μεσαίο είναι μικρότερο*

Left ← M+1 *! ξεκινά η αναζήτηση από το επόμενο του μέσου*

Αλλιώς

Right ← M-1 *! τέλος αναζήτησης γίνεται ο προηγούμενος του μέσου*

Τέλος_αν

Τέλος_επανάληψης

Αν Flag = αληθής τότε

Εμφάνισε "Το στοιχείο,", key , "υπάρχει στη θέση:", pos

Αλλιώς

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

Τέλος_αν

① ① ① ① ① ① ① Αναζήτηση χρησιμοποιούμε όταν από την εκφώνηση ζητείται:

- να διαβάσουμε ένα δεδομένο και για αυτό να βρούμε κάποια στοιχεία που υπάρχουν σε έναν ή περισσότερους πίνακες (διαβάζουμε AT και να εμφανίσουμε όνομα)

- Ζητείται για ένα δεδομένο να βρούμε κάποια στοιχεία που υπάρχουν σε ένα ή περισσότερους πίνακες τα οποία σχετίζονται με αυτό το δεδομένο (να βρούμε ο μαθητής Παπαδόπουλος ποιο μέσο όρο έχει)

Λυμένα Παραδείγματα

14. Ένα λογιστικό γραφείο διατηρεί μια λίστα με ονοματεπώνυμα πελατών και χρήματα σε € που χρωστάει καθένας από τους 1200 πελάτες του γραφείου. Ο ιδιοκτήτης του γραφείου καταχωρίζει τα στοιχεία των πελατών του σε μια μία ή περισσότερες δομές δεδομένων με τη βοήθεια προγράμματος. Έπειτα θέλει να εμφανίζει: α) το ονοματεπώνυμο του πελάτη ή των πελατών που χρωστάνε τα περισσότερα χρήματα, β) τα χρήματα που χρωστά κάποιος πελάτης αν δώσει ως είσοδο στον αλγόριθμο ένα ονοματεπώνυμο. Αν το ονοματεπώνυμο δεν αντιστοιχεί σε πελάτη να εκτυπώνεται ότι ο πελάτης δεν υπάρχει.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I
    ΠΡΑΓΜΑΤΙΚΕΣ: Χ[1200], ΜΑΧ
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[1200], ΟΝΟΜΑ
    ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 1200
    ΔΙΑΒΑΣΕ Ο[I], Χ[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ ← -1
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 1200
    ΑΝ Χ[I] > ΜΑΧ ΤΟΤΕ
        ΜΑΧ ← Χ[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 1200
    ΑΝ Χ[I]= ΜΑΧ ΤΟΤΕ
        ΓΡΑΨΕ Ο[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ ΟΝΟΜΑ
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
I ← 1
ΟΣΟ I <=1200 ΚΑΙ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ Ο[I] = ΟΝΟΜΑ ΤΟΤΕ
        ΓΡΑΨΕ Χ[I]
        ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ
        I ← I + 1
    ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Ο ΠΕΛΑΤΗΣ ΔΕΝ ΒΡΕΘΗΚΕ ΣΤΟΝ ΠΙΝΑΚΑ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

15. Η παιδική κατασκήνωση ΠΗΓΑΣΟΣ έχει 25 σκηνές στην κάθε μια από τις οποίες μένει διαφορετικός αριθμός παιδιών. Όταν ένα παιδί έρχεται για πρώτη φορά στην κατασκήνωση, η γραμματεία καταχωρίζει το επίθετο, το όνομα, τη διεύθυνση και τον αριθμό σκηνής που μένει. Αν κατά το πρώτο 15μερο του Ιουλίου υπάρχουν συνολικά σε

όλες τις σκηνές 250 παιδιά να γραφεί πρόγραμμα που α. θα εισάγει τα απαραίτητα στοιχεία του καθενός σε κατάλληλους πίνακες. β. θα διαβάσει ένα αριθμό σκηνής, εξασφαλίζοντας ότι είναι από το 1 μέχρι το 25 και να εμφανίζει τα στοιχεία των παιδιών που μένουν στη σκηνή και στο συνολικό αριθμό αυτών. γ. θα διαβάσει το επίθετο και το όνομα ενός παιδιού και θα εμφανίζει τον αριθμό της σκηνής που μένει.

ΠΡΟΓΡΑΜΜΑ ΚΑΤΑΣΚΗΝΩΣΗ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: I, A_Σ [250], ΣΚΗΝΗ, ΠΛΗΘΟΣ

ΧΑΡΑΚΤΗΡΕΣ: E[250], O[250], Δ[250], ΕΠΙΘΕΤΟ, ΟΝΟΜΑ

ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ

ΑΡΧΗ

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 250

ΔΙΑΒΑΣΕ E[I], O[I], Δ[I], A_Σ[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ΣΚΗΝΗ

ΜΕΧΡΙΣ_ΟΤΟΥ ΣΚΗΝΗ > = 1 ΚΑΙ ΣΚΗΝΗ <=25

ΠΛΗΘΟΣ ← 0

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 250

ΑΝ A_Σ[I] = ΣΚΗΝΗ ΤΟΤΕ

ΠΛΗΘΟΣ ← ΠΛΗΘΟΣ + 1

ΓΡΑΨΕ E[I], O[I], Δ[I]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ΠΛΗΘΟΣ

ΔΙΑΒΑΣΕ ΕΠΙΘΕΤΟ, ΟΝΟΜΑ

ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ

I ← 1

ΟΣΟ FLAG = ΨΕΥΔΗΣ ΚΑΙ I <=250 ΕΠΑΝΑΛΑΒΕ

ΑΝ O[I] = ΟΝΟΜΑ ΚΑΙ E[I] = ΕΠΙΘΕΤΟ ΤΟΤΕ

ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ

ΓΡΑΨΕ A_Σ[I]

ΑΛΛΙΩΣ

I ← I + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ ΤΟ ΠΑΙΔΙ'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΤΑΣΚΗΝΩΣΗ

Ασκήσεις

13. Να γράψετε πρόγραμμα που α. για κάθε ένα από τα δέκα μουσεία που υπάρχουν, να διαβάσει και να καταχωρίζει σε μονοδιάστατους πίνακες, την ονομασία του, τη διεύθυνσή του, το τηλέφωνο του, τις ημέρες λειτουργίας του. Για τις ημέρες λειτουργίας θα εισάγεται το γράμμα 'Π' αν λειτουργεί πενθήμερο, το 'Σ' όταν λειτουργεί πενθήμερο και το Σάββατο και το 'Ε' όταν λειτουργεί όλη την εβδομάδα. β. Θα διαβάσει την ονομασία ενός μουσείου και να εμφανίζει την διεύθυνση του, το τηλέφωνό του και ανάλογα τις μέρες λειτουργίας τις λέξεις 'Πενθήμερο', ή 'Πενθήμερο και Σάββατο' ή 'Όλη την εβδομάδα'. γ. Θα εμφανίζει τις πληροφορίες όλων των μουσείων που λειτουργούν μόνο το 'Πενθήμερο'.

14. Για να μπορέσει κάποιος να πραγματοποιήσει αγορές από ένα ηλεκτρονικό κατάστημα θα πρέπει να έχει εγγραφεί σε αυτό. Πριν την έναρξη των αγορών πραγματοποιείται ταυτοποίηση του πελάτη με την εξής διαδικασία: ο διαχειριστής έχει καταχωρημένα στον πίνακα ΟΝ το όνομα του χρήστη και στον πίνακα ΚΩΔ τον κωδικό

πρόσβασης των 1500 ατόμων που έχουν γραφτεί στο κατάστημα. Στην ιστοσελίδα ο πελάτης εισάγει το όνομα χρήστη και τον κωδικό και αν αυτά υπάρχουν στους πίνακες του διαχειριστή μπορεί να κάνει αγορές. Να γραφεί πρόγραμμα που: α. Θα εισάγει στον πίνακα ON το όνομα χρήστη και στον πίνακα ΚΩΔ τον κωδικό πρόσβασης των 1500 πελατών του καταστήματος., β. Θα πραγματοποιεί ταυτοποίηση ως εξής: θα διαβάσει το όνομα χρήστη και τον κωδικό πρόσβασης. Μετά την εισαγωγή, αν το όνομα χρήστη δεν υπάρχει καταχωρισμένο στον πίνακα ON θα εμφανίζει το μήνυμα 'Ο χρήστης δεν υπάρχει' και θα τερματίζει. Αν το όνομα βρεθεί τότε θα εξετάζει αν ο κωδικός υπάρχει στον πίνακα ΚΩΔ και αν δεν υπάρχει να εμφανίζει το μήνυμα 'Λάθος κωδικός' και να τερματίζει. Αν το όνομα και κωδικός βρεθούν τότε να προχωρά στις αγορές. Μετά την επιτυχή ταυτοποίηση του πελάτη, να διαβάσει συνεχώς το είδος και την τιμή ενός αγαθού που θέλει να αγοράσει μέχρι να δοθεί η τελεία ως είδος αγαθού. Να εμφανίζει το συνολικό ποσό των αγορών που πραγματοποίησε ο πελάτης.
Παρατήρηση: Ο κωδικός αποτελείται από γράμματα και αριθμούς.

15. Στον πίνακα Χ[28] υπάρχουν αποθηκευμένα τα ονόματα 28 χωρών της ΕΕ και στον πίνακα Π[28] ο αντίστοιχος πληθυσμός της καθεμίας. Να γραφεί πρόγραμμα που: α. για κάθε χώρα θα διαβάσει το όνομα και τον πληθυσμό της και να αποθηκεύει στους πίνακες Χ και Π αντίστοιχα. β. για κάθε μια χώρα θα διαβάσει τις ονομασίες των τεσσάρων μεγαλύτερων πόλεων της και να τις καταχωρίζει στον πίνακα ON[112], τον πληθυσμό της κάθε μιας από τις πόλεις και να τις καταχωρίζει στον πίνακα ΠΛ[112], τη χώρα στην οποία ανήκει η κάθε πόλη και να την καταχωρίζει στον πίνακα ΧΩΡ[112]. γ. Να εμφανίζει τις ονομασίες των χωρών στις οποίες πάνω από το 40% του πληθυσμού τους κατοικεί σε μια μόνο πόλη τους. Αν δεν βρεθεί τέτοια χώρα να εμφανίζει το μήνυμα «Δεν υπάρχουν πόλεις».

Ταξινόμηση πίνακα

Γενικά

Η ταξινόμηση σε μια δομή δεδομένων έχει ως στόχο τη διάταξη των δεδομένων με μια ιδιαίτερη σειρά. Η σειρά αυτή μπορεί να είναι τέτοια που να τοποθετούνται τα δεδομένα από το μικρότερο στο μεγαλύτερο (**αύξουσα**) ή από το μεγαλύτερο στο μικρότερο (**φθίνουσα**).

Χρησιμοποιείται σε περιπτώσεις αναζήτησης αλφαβητικών και αριθμητικών δεδομένων σε καταλόγους, λεξικά κλπ.

Αύξουσα: Δοθέντων των στοιχείων a_1, a_2, \dots, a_n η ταξινόμηση συνίσταται στη μετάθεση της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μια σειρά $a_{k1}, a_{k2}, \dots, a_{kn}$ ώστε δοθείσης μιας συνάρτησης διάταξης f , να ισχύει $f(a_{k1}) \leq f(a_{k2}) \leq \dots \leq f(a_{kn})$

Στην **φθίνουσα** ισχύσει $f(a_{k1}) \geq f(a_{k2}) \geq \dots \geq f(a_{kn})$

Μέθοδος της φυσαλίδας μονοδιάστατου πίνακα

Η ταξινόμηση των στοιχείων a_1, a_2, \dots γίνεται με τη μετάθεση της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μια σειρά.

Θα δούμε τη **μέθοδο της φυσαλίδας (Bubble Sort) ή ευθείας ανταλλαγής**. Είναι ο πιο απλός και πιο αργός αλγόριθμος ταξινόμησης.

Άλλοι απλοί αλγόριθμοι είναι η ταξινόμηση με ευθεία επιλογή και η ταξινόμηση με ευθεία εισαγωγή.

Ο αλγόριθμος φυσαλίδας βασίζεται στις διαδοχικές αντιμεταθέσεις στοιχείων έτσι ώστε τα στοιχεία με τις μικρότερες τιμές να ανέβουν στις πρώτες θέσεις του πίνακα. 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) μετακινείται προς την κορυφή του πίνακα. Αφού γίνουν όλες οι σαρώσεις θα έχει επιτευχθεί η ταξινόμηση.

Αλγόριθμος Φυσαλίδα

Δεδομένα //N, Π //

Για i από 2 μέχρι N

 Για j από N μέχρι i με_βήμα -1

 Αν $\Pi[j-1] > \Pi[j]$ τότε

 temp \leftarrow $\Pi[j-1]$

$\Pi[j-1] \leftarrow \Pi[j]$

$\Pi[j] \leftarrow$ temp

 τέλος_αν

 τέλος_επανάληψης

 τέλος_επανάληψης

Αποτελέσματα //Π//

Τέλος Φυσαλίδα

!το i σαρώνει από τη $2^{\text{η}}$ θέση έως το τέλος

!το j σαρώνει από το τέλος προς τη θέση i

!αν το $j-1$ στοιχείο είναι μεγαλύτερο από το j στοιχείο τότε

!κάνε αντιμετάθεση στοιχείων. Δηλαδή, το j στοιχείο θα πάει στη

θέση του $j-1$ και το $j-1$ στη θέση του j . Επειδή δεν μπορεί να γίνει άμεσα

!διότι θα χαθεί το στοιχείο $j-1$, προσωρινά τοποθετείται στη βοηθητική

!μεταβλητή temp.

BBB Αντί των εντολών:

 temp \leftarrow $\Pi[j-1]$

$\Pi[j-1] \leftarrow \Pi[j]$

$\Pi[j] \leftarrow$ temp

Μπορούμε να χρησιμοποιήσουμε την εντολή **Αντιμετάθεση $\Pi[j-1], \Pi[j]$** (μόνο σε αλγόριθμο)

Παράδειγμα : Να ταξινομηθούν οι ακέραιοι 12, 7, 5, 3, 10 ενός πίνακα. Πώς διαμορφώνεται ο πίνακας σε κάθε βήμα του αλγορίθμου;

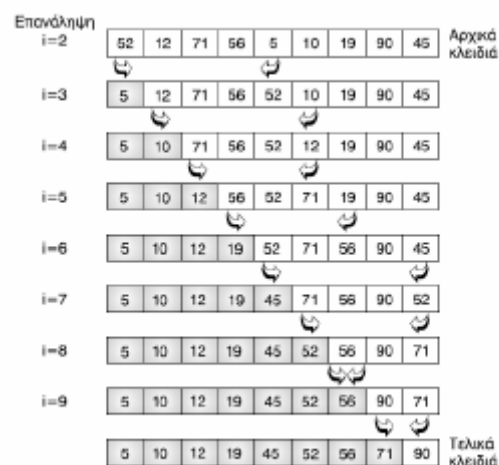
Θέσεις πίνακα				
1	2	3	4	5
12	7	5	3	10
12	7	3	5	10
12	3	7	5	10
3	12	7	5	10
3	12	7	5	10
3	12	5	7	10
3	5	12	7	10
3	5	12	7	10
3	5	7	12	10
3	5	7	10	12

Ταξινόμηση με ευθεία επιλογή μονοδιάστατου πίνακα

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

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

Ο Αλγόριθμος ταξινόμησης με επιλογή είναι ο παρακάτω.



Σχήμα 3.2. Ταξινόμηση ευθείας επιλογής.

Αλγόριθμος Ταξινόμηση με ευθεία επιλογή

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

!η εξωτερική επανάληψη δίνει το στοιχείο που συγκρίνεται

!κρατάμε τη θέση του

!εξετάζουμε τα υπόλοιπα στοιχεία

!(αύξουσα) αν κάποιο είναι μικρότερο κρατάμε τη θέση του

!αντιμεταθέτουμε το μικρότερο με αυτό που συγκρίνεται

Λυμένα Παραδείγματα

16. Να γραφεί πρόγραμμα που θα διαβάσει έναν πίνακα 50 θέσεων και θα εμφανίζει τους 5 μεγαλύτερους αριθμούς του πίνακα.

```
ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, J
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[50], TEMP1
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 50
    ΔΙΑΒΑΣΕ Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 50
    ΓΙΑ J ΑΠΟ 50 ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
        ΑΝ Π[J-1] < Π[J] ΤΟΤΕ
            TEMP1 ← Π[J-1]
            Π[J-1] ← Π[J]
            Π[J] ← TEMP1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΡΑΨΕ Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ
```

17. Σε τρεις πίνακες T, E και M 220 θέσεων αποθηκεύουμε την τάξη («Α» ή «Β» ή «Γ»), το επίθετο και το μέσο όρο των μαθητών ενός σχολείου. Να εμφανιστούν οι μαθητές της Α ταξινομημένοι σε φθίνουσα σειρά βαθμολογίας. Σε περίπτωση ισοβαθμίας μεταξύ των μαθητών της Α να εμφανίζονται πρώτα οι μαθητές με το μικρότερο αλφαβητικά επίθετο.

```
ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, J
    ΧΑΡΑΚΤΗΡΕΣ: T[220], E[220], TEMP1, TEMP2
    ΠΡΑΓΜΑΤΙΚΕΣ: M[220], TEMP3
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 220
    ΔΙΑΒΑΣΕ T[I], E[I], M[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 220
    ΓΙΑ J ΑΠΟ 220 ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
        ΑΝ M[J-1] < M[J] ΤΟΤΕ
            TEMP1 ← T[J-1]
            T[J-1] ← T[J]
            T[J] ← TEMP1
            TEMP2 ← E[J-1]
            E[J-1] ← E[J]
            E[J] ← TEMP2
            TEMP3 ← M[J-1]
            M[J-1] ← M[J]
            M[J] ← TEMP3
        ΑΛΛΙΩΣ ΑΝ M[J-1] = M[J] ΤΟΤΕ
            ΑΝ E[J-1] > E[J] ΤΟΤΕ
                TEMP1 ← T[J-1]
                T[J-1] ← T[J]
                T[J] ← TEMP1
```

```

TEMP2 ← E[J-1]
E[J-1] ← E[J]
E[J] ← TEMP2
TEΛΟΣ_ΑΝ
TEΛΟΣ_ΑΝ
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 220
ΑΝ T[I] = 'Α' ΤΟΤΕ
ΓΡΑΨΕ E[I], M[I]
TEΛΟΣ_ΑΝ
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
TEΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

18. Ένας διαγωνισμός τραγουδιού στην Ευρώπη διεξάγεται ως εξής: Γίνεται μια πρώτη ακρόαση του τραγουδιού κάθε μιας από τις 25 συμμετέχουσες χώρες από την κριτική επιτροπή η οποία δίνει βαθμούς από 1-100. Με πρόγραμμα να διαβαστεί το όνομα κάθε χώρας και η βαθμολογία της. Στη συνέχεια θα επιλέγει για την δεύτερη φάση του διαγωνισμού τις χώρες με τη μεγαλύτερη βαθμολογία. Πρέπει όμως το άθροισμα βαθμολογίας των χωρών που θα προχωρήσουν στη 2^η φάση να είναι μικρότερο ή ίσο των 1000 βαθμών. Να εμφανιστούν ταξινομημένα αλφαβητικά τα ονόματα των χωρών που περνάνε στη 2^η φάση.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: I, J, K,
ΧΑΡΑΚΤΗΡΕΣ: O[25], TEMP2
ΠΡΑΓΜΑΤΙΚΕΣ: B[25], TEMP1, A
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 25
ΔΙΑΒΑΣΕ O[I], B[I]
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 25
ΓΙΑ J ΑΠΟ 25 ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
ΑΝ B[J-1] < B[J] ΤΟΤΕ
TEMP1 ← B[J-1]
B[J-1] ← B[J]
B[J] ← TEMP1
TEMP2 ← O[J-1]
O[J-1] ← O[J]
O[J] ← TEMP2
TEΛΟΣ_ΑΝ
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
A ← 0
K ← 0
I ← 1
ΟΣΟ A + B [I] <= 1000 ΚΑΙ I <= 25 ΕΠΑΝΑΛΑΒΕ
A ← A + B[I]
K ← K + 1
I ← I + 1
TEΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ K
ΓΙΑ J ΑΠΟ K ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
ΑΝ O[J-1] > O[J] ΤΟΤΕ
TEMP1 ← B[J-1]
B[J-1] ← B[J]
B[J] ← TEMP1
TEMP2 ← O[J-1]

```

```

        O[J-1] ← O[J]
        O[J] ← TEMP2
    ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Κ
    ΓΡΑΨΕ O[I], B[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

19. Για τη στελέχωση μιας ιατρικής ομάδας ενός ερευνητικού ιδρύματος, προσήλθε μεγάλος αριθμός υποψηφίων που ο καθένας υποβλήθηκε σε 16 δοκιμασίες και βαθμολογήθηκε ξεχωριστά για την κάθε μια με ένα βαθμό από 1 έως 100. Ο τελικός βαθμός προήλθε από το μέσο όρο των 5 καλύτερων βαθμών που έλαβε. Να αναπτύξετε πρόγραμμα που α. διαβάζει το ονοματεπώνυμο ενός υποψηφίου, β. να διαβάζει τους βαθμούς τους στις 16 δοκιμασίες και να τις καταχωρίζει στον πίνακα Β, γ. να βρίσκει τον τελικό βαθμό του υποψηφίου. Τα παραπάνω να επαναλαμβάνονται μέχρις ότου να δοθεί ως ονοματεπώνυμο η λέξη «ΤΕΛΟΣ». δ. Να εμφανίζει το ονοματεπώνυμο του υποψηφίου με τον μεγαλύτερο τελικό βαθμό. Να λάβετε υπόψη την περίπτωση κατά την οποία μπορεί να μην έχει διαβαστεί κανείς υποψήφιος. Θεωρείστε ότι δεν υπάρχουν ίδιοι τελικοί βαθμοί και ότι το ονοματεπώνυμο γράφεται με κεφαλαία γράμματα της ελληνικής αλφαβήτου.

```

ΠΡΟΓΡΑΜΜΑ ΕΡΕΥΝΗΤΙΚΗ_ΟΜΑΔΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, J, B[16], TEMP1
    ΠΡΑΓΜΑΤΙΚΕΣ: ΤΕΛ_ΒΑΘ, MAX
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝΟΜΑ, ΟΝΟΜΑ_MAX
ΑΡΧΗ
MAX ← 0
ΔΙΑΒΑΣΕ ΟΝΟΜΑ
ΟΣΟ ΟΝΟΜΑ <> 'ΤΕΛΟΣ' ΕΠΑΝΑΛΑΒΕ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 16
        ΔΙΑΒΑΣΕ B[I]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 16
            ΓΙΑ J ΑΠΟ 16 ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
                ΑΝ B[J-1] < B[J] ΤΟΤΕ
                    TEMP1 ← B[J-1]
                    B[J-1] ← B[J]
                    B[J] ← TEMP1
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛ_ΒΑΘ ← (B[1] + B[2] + B[3] + B[4] + B[5]) / 5
ΑΝ ΤΕΛ_ΒΑΘ > MAX ΤΟΤΕ
    MAX ← ΤΕΛ_ΒΑΘ
    ΟΝΟΜΑ_MAX ← ΟΝΟΜΑ
ΤΕΛΟΣ_ΑΝ
ΔΙΑΒΑΣΕ ΟΝΟΜΑ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ MAX = 0 ΤΟΤΕ
    ΓΡΑΨΕ 'ΔΕΝ ΠΡΟΣΗΛΘΕ ΚΑΝΕΙΣ ΥΠΟΨΗΦΙΟΣ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ ΟΝΟΜΑ_MAX
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΕΡΕΥΝΗΤΙΚΗ_ΟΜΑΔΑ

```

20. Ένας μαθητής έχει μια συλλογή από 150 δίσκους. Για κάθε δίσκο έχει καταγράψει στον πίνακα T τον τίτλο και στον πίνακα X τη χρονιά έκδοσής του. Να γραφεί πρόγραμμα που α. για κάθε ένα από τους 150 δίσκους να διαβάζει τον τίτλο και τη χρονιά έκδοσης και να τα αποθηκεύει στους πίνακες T και X. β. Ταξινομεί τους δίσκους με βάση τη χρονιά έκδοσής κατά αύξουσα σειρά. γ. Να διαβάζει τον τίτλο ενός δίσκου και εάν αυτό υπάρχει στη συλλογή να εμφανίζει τη χρονιά έκδοσής του διαφορετικά να εμφανίζει το μήνυμα «Δεν βρέθηκε».

```

ΠΡΟΓΡΑΜΜΑ ΣΥΛΛΟΓΗ_CD
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, J, Χ[150], TEMP1, ΑΡΧΙΚΟΣ, ΤΕΛΙΚΟΣ, ΜΕΣΟΣ, ΘΕΣΗ
    ΧΑΡΑΚΤΗΡΕΣ: T [150], TEMP2, ΤΙΤΛΟΣ
    ΛΟΓΙΚΕΣ: FLAG
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 150
    ΔΙΑΒΑΣΕ T [I], Χ[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 150
    ΓΙΑ J ΑΠΟ 150 ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
        ΑΝ Χ[J-1]> Χ[J] ΤΟΤΕ
            TEMP1 ← Χ[J-1]
            Χ[J-1] ← Χ[J]
            Χ[J] ← TEMP1
            TEMP2 ← T[J-1]
            T[J-1] ← T[J]
            T[J]← TEMP2
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
FLAG ← ΨΕΥΔΗΣ
ΑΡΧΙΚΟΣ ← 1
ΤΕΛΙΚΟΣ ← 150
ΔΙΑΒΑΣΕ ΤΙΤΛΟΣ
ΟΣΟ FLAG = ΨΕΥΔΗΣ ΚΑΙ ΑΡΧΙΚΟΣ <=ΤΕΛΙΚΟΣ ΕΠΑΝΑΛΑΒΕ
    ΜΕΣΟΣ ← (ΑΡΧΙΚΟΣ + ΤΕΛΙΚΟΣ) DIV 2
    ΑΝ T [ΜΕΣΟΣ] = ΤΙΤΛΟΣ ΤΟΤΕ
        FLAG ← ΑΛΗΘΗΣ
        ΘΕΣΗ ←ΜΕΣΟΣ
    ΑΛΛΙΩΣ ΑΝ T [ΜΕΣΟΣ] < ΤΙΤΛΟΣ ΤΟΤΕ
        ΑΡΧΙΚΟΣ ←ΜΕΣΟΣ +1
    ΑΛΛΙΩΣ
        ΤΕΛΙΚΟΣ ←ΜΕΣΟΣ -1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ FLAG = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ Χ[ΘΕΣΗ]
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΤΟ CD ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΥΛΛΟΓΗ_CD

```

21. Να γραφεί πρόγραμμα σε γλώσσα που θα υλοποιεί το παρακάτω πρόγραμμα: α. Για κάθε ένα από τα 500 βιβλία μιας αποθήκης βιβλίων, θα διαβάζει τον τίτλο του, τον συγγραφέα και τον αριθμό των αντιτύπων. β. Θα ταξινομεί τα βιβλία προς τον αριθμό αντιτύπων κατά αύξουσα σειρά. γ. Θα αναζητεί και θα εμφανίζει τους τίτλους και τους συγγραφείς των βιβλίων των οποίων υπάρχουν μόνο 5 αντίτυπα στην αποθήκη. Η αναζήτηση να διακόπτεται μόλις εμφανιστούν όλοι οι τίτλοι και οι συγγραφείς. Θεωρείστε ότι υπάρχει σίγουρα έστω και ένα τέτοιο βιβλίο.

```

ΠΡΟΓΡΑΜΜΑ ΑΠΟΘΗΚΗ_ΒΙΒΛΙΩΝ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, J, ΑΝΤ[500], TEMP1
    ΧΑΡΑΚΤΗΡΕΣ: ΤΙΤ[500], ΣΥΓ[250], TEMP2, TEMP3
    ΛΟΓΙΚΕΣ: FLAG
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 500
    ΔΙΑΒΑΣΕ ΤΙΤ[I], ΣΥΓ[I], ΑΝΤ[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 500
    ΓΙΑ J ΑΠΟ 500 ΜΕΧΡΙ I ΜΕ ΒΗΜΑ -1
        ΑΝ ΑΝΤ[J-1]> ΑΝΤ[J] ΤΟΤΕ
            TEMP1 ← ΑΝΤ[J-1]
            ΑΝΤ[J-1] ← ΑΝΤ[J]
            ΑΝΤ[J] ← TEMP1
            TEMP2 ← ΤΙΤ[J-1]
            ΤΙΤ [J-1] ← ΤΙΤ [J]
            ΤΙΤ [J]← TEMP2
            TEMP3 ←ΣΥΓ[J-1]
            ΣΥΓ[J-1]← ΣΥΓ[J]
            ΣΥΓ [J]← TEMP3
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
FLAG ← ΨΕΥΔΗΣ
I ← 1
ΟΣΟ FLAG = ΨΕΥΔΗΣ ΚΑΙ I <=500 ΕΠΑΝΑΛΑΒΕ
    ΑΝ ΑΝΤ [I] = 5 ΤΟΤΕ
        ΓΡΑΨΕ ΤΙΤ[I], ΣΥΓ[I]
    ΑΛΛΙΩΣ ΑΝ ΑΝΤ [I] > 5 ΤΟΤΕ
        FLAG ← ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
    I ← I + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΠΟΘΗΚΗ_ΒΙΒΛΙΩΝ

```

22. Με δεδομένους 2 ταξινομημένους σε αύξουσα σειρά πίνακες N και M ακεραίων αντίστοιχα να γραφεί τμήμα προγράμματος που θα κάνει συγχώνευση των πινάκων σε νέο πίνακα.

```

I←1
J←1
K←1
ΟΣΟ I<=N ΚΑΙ J<=M ΕΠΑΝΑΛΑΒΕ
    ΑΝ A[I]<B[J] ΤΟΤΕ
        Γ[K]←A[I]
        K←K+1
        I←I+1
    ΑΛΛΙΩΣ
        Γ[K]←B[J]
        K←K+1
        J←J+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ I>N ΤΟΤΕ
    ΓΙΑ Λ ΑΠΟ K ΜΕΧΡΙ N+M
        Γ[Λ]←B[J]
        J←J+1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΑΛΛΙΩΣ
  ΓΙΑ Λ ΑΠ'Ο Κ ΜΕΧΡΙ Ν+Μ
    Γ[Λ]←Α[Ι]
    Ι←Ι+1
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ

```

ΕΝΑΛΛΑΚΤΙΚΟΣ ΤΡΟΠΟΣ

```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  Γ[Ι]←Α[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Μ
  Γ[Ι+Ν]←Β[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν+Μ
  ΓΙΑ J ΑΠ'Ο Ν+Μ ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
    ΑΝ Γ[J-1]> Γ[J] ΤΟΤΕ
      ΑΝΤΙΜΕΤΑΘΕΣΕ Γ[J-1], Γ[J]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Ασκήσεις

16. Στο τμήμα πληροφορικής εισηχθησαν 120 φοιτητές. Να αναπτύξετε πρόγραμμα που:

- για κάθε φοιτητή θα διαβάζει το όνομα και τα μόρια που συγκέντρωσε και θα καταχωρίζει στους πίνακες Ον και ΜΟ, β. να εκτυπώνει τα ονόματα των φοιτητών δίπλα στα μόριά τους, ταξινομημένα με βάση τα μόρια κατά φθίνουσα σειρά. Σε περίπτωση που βρεθούν φοιτητές με ίδια μόρια, η σειρά ταξινόμησης των ονομάτων να είναι αλφαβητική.

17. Στο πρωτάθλημα ποδοσφαίρου μια χώρας συμμετέχουν 18 ομάδες που αγωνίζονται μεταξύ τους συγκεντρώνοντας βαθμούς για να προκύψει στο τέλος του πρωταθλήματος η τελική βαθμολογία. Οι 6 πρώτοι συμμετέχουν στις ευρωπαϊκές διοργανώσεις ως εξής: ο πρώτος στους ομίλους του Τσάμπιονς Λίγκ, ο δεύτερος στα προκριματικά του Τσάμπιονς Λίγκ, ο τρίτος, τέταρτος και πέμπτος στο Europa League και ο έκτος στο κύπελλο Ιντερτότο. Να αναπτύξετε πρόγραμμα που:

- θα διαβάζει τα ονόματα των ομάδων του πρωταθλήματος και θα τα καταχωρίζει στο πίνακα ΟΝ, β. για κάθε ομάδα θα διαβάζει τον αριθμό των νικών, των ισοπαλιών και θα τα καταχωρίζει στους πίνακες Ν και ΙΣ αντίστοιχα, γ. Με δεδομένο ότι με την νίκη η ομάδα παίρνει 3 βαθμούς και για κάθε ισοπαλία 1 βαθμό, να υπολογίζει και να καταχωρίζει στον πίνακα ΒΑΘ τη συνολική βαθμολογία της ομάδας, δ. να εμφανίζει τα ονόματα των ομάδων που θα συμμετάσχουν στις ευρωπαϊκές διοργανώσεις και ποιες διοργανώσεις θα είναι αυτές

18. Ένας αγώνας ποδηλατοδρομίας διήρκησε δύο μέρες: κατά την πρώτη μέρα οι 64 συμμετέχοντες χωρίστηκαν σε δύο γκρουπ των 32 αθλητών το καθένα και πραγματοποίησαν μια συγκεκριμένη διαδρομή. Οι δεκαέξι καλύτεροι αθλητές (αυτοί που χρειάστηκαν το λιγότερο χρόνο) και από τα δύο γκρουπ προκρίνονται στον τελικό που θα διεξαχθεί την δεύτερη μέρα των αγώνων. Να γραφεί πρόγραμμα σε γλώσσα που:

- θα διαβάζει τα ονόματα των 32 αθλητών του πρώτου γκρουπ και τους χρόνους τους σε δευτερόλεπτα (ακέραιες τιμές) και να τους καταχωρίζει στους πίνακες ΟΝ1, ΧΡ1 αντίστοιχα, β. θα διαβάζει τα ονόματα των 32 αθλητών του δεύτερου γκρουπ και τους χρόνους τους σε δευτερόλεπτα (ακέραιες τιμές) και να τους καταχωρίζει στους πίνακες ΟΝ2, ΧΡ2 αντίστοιχα, γ. θα δημιουργεί τους πίνακες ΟΝ και ΧΡ αντίστοιχα που θα περιέχουν τα ονόματα και τους χρόνους των αθλητών που θα συμμετάσχουν στον τελικό,

δ. θα εμφανίζει τα ονόματα δίπλα στους χρόνους των αθλητών του τελικού ξεκινώντας από αυτόν με τον λιγότερο χρόνο.

Παρατήρηση: δεν υπάρχουν αθλητές με το ίδιο χρόνο.

19. Ο κάθε αθλητής που συμμετείχε στους αγώνες ιστιοσανίδας πραγματοποίησε 12 διαδρομές και για κάθε μια διαδρομή λαμβάνει από μια επιτροπή ένα βαθμό από 1 έως το 20 (ακέραιος αριθμός). Από τις 12 συνολικά βαθμολογίες αφαιρούνται οι 3 χαμηλότερες και το άθροισμα των υπολοίπων δημιουργεί την τελική βαθμολογία. Να αναπτύξετε πρόγραμμα που: α. Για κάθε αθλητή να διαβάζει το ονοματεπώνυμό του, να διαβάζει την βαθμολογία που έλαβε σε κάθε μια από τις 12 διαδρομές και να υπολογίζει την τελική βαθμολογία του. Το τέλος του διαβάσματος να γίνει όταν δοθεί η λέξη ΤΕΛΟΣ ως ονοματεπώνυμο του αθλητή. β. Να εμφανίζει το όνομα του αθλητή με την υψηλότερη βαθμολογία.

20. Να αναπτύξετε πρόγραμμα που: α) να διαβάζει τις επωνυμίες, τα ετήσια έσοδα και τα ετήσια έξοδα 100 επιχειρήσεων και αποθηκεύει σε τρεις μονοδιάστατους πίνακες αντίστοιχα, β) να ταξινομεί τις επιχειρήσεις ως προς τα κέρδη τους, γ) να εμφανίζει τα ονόματα και τα έσοδα των δέκα επιχειρήσεων με τα μεγαλύτερα έσοδα, δ) να αναζητά και να εμφανίζει τα ονόματα και το πλήθος των επιχειρήσεων με ζημίες, ε) τέλος, να εμφανίζει τα ονόματα και το πλήθος των επιχειρήσεων με κέρδη άνω των 1.000.000 € καθώς και το όνομα της επιχείρησης με τα μικρότερα κέρδη.

Δισδιάστατοι πίνακες (πολλών γραμμών και στηλών)

Είναι ουσιαστικά μια μεταβλητή στην οποία κάθε χρονική στιγμή είναι αποθηκευμένες πολλές τιμές. Οι τιμές αυτές είναι διατεταγμένες σε δύο διαστάσεις, οριζοντίως και καθέτως, σε γραμμές και στήλες αντίστοιχα.

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

Στο παρακάτω παράδειγμα ο πίνακας ορίζεται ως εξής : ακεραίος $\Pi [3, 4]$.

Η πρώτη διάσταση αναφέρεται στις γραμμές και η δεύτερη στις στήλες. Γενικά, ορίζεται ως $\Pi [M, N]$ για έναν πίνακα $M \times N$.

	1	2	3	4	
1	$\Pi[1,1]$	$\Pi[1,2]$	$\Pi[1,3]$	$\Pi[1,4]$	1
2	$\Pi[2,1]$	$\Pi[2,2]$	$\Pi[2,3]$	$\Pi[2,4]$	2
3	$\Pi[3,1]$	$\Pi[3,2]$	$\Pi[3,3]$	$\Pi[3,3]$	3

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

Οι πίνακες με ίσο αριθμό γραμμών και στηλών ονομάζονται τετραγωνικοί. Τα στοιχεία για τα οποία ισχύει γραμμή=στήλη ($i=j$) βρίσκονται στην κύρια διαγώνιο. Η άλλη διαγώνιος λέγεται δευτερεύουσα.

- Για τα στοιχεία που αποτελούν την κύρια διαγώνιο ισχύει: $i=j$
- Για τα στοιχεία που αποτελούν τη δευτερεύουσα διαγώνιο ισχύει: $i+j=N+1$
- Για τα στοιχεία που βρίσκονται πάνω από την κύρια διαγώνιο ισχύει: $j>i$
- Για τα στοιχεία που βρίσκονται κάτω από την κύρια διαγώνιο ισχύει: $j<i$

Λυμένα Παραδείγματα

23. Έστω Λ πίνακας 100×200 που περιέχει λέξεις και K πίνακας 10×30 που περιέχει τις τιμές αληθής και ψευδής. Να γραφεί η δήλωση των πινάκων

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: $\Lambda [100,200]$

ΛΟΓΙΚΕΣ: $K [10,30]$

24. Δίνεται ο διπλανός δισδιάστατος πίνακας A . Να εξετάσετε ποιες εντολές από τις παρακάτω είναι σωστές και ποιες όχι. Δικαιολογήστε την απάντησή σας.

0	-5	2	1
4	6	7	0
2	-6	8	9
-7	3	4	-5

α. $A[3,3] \leftarrow 10$
δ. $A[2.5,3] \leftarrow 20$

β. $A[6,4] \leftarrow 12$
ε. ΓΡΑΨΕ $A[(6 \text{ div } 3),4]$

γ. $A[2,3]+1 \leftarrow 5$
στ. $A[2] \leftarrow 15$

Α. ΣΩΣΤΗ. ΤΟΠΟΘΕΤΕΙ ΤΟΝ ΑΡΙΘΜΟ 10 ΣΤΟ ΣΤΟΙΧΕΙΟ $A[3,3]$.

Β. ΛΑΝΘΑΣΜΕΝΗ, ΓΙΑΤΙ ΠΡΟΣΠΑΘΕΙ ΝΑ ΕΚΧΩΡΗΣΕΙ ΤΟΝ ΑΡΙΘΜΟ 12 ΣΕ ΕΝΑ ΣΤΟΙΧΕΙΟ ΤΟΥ ΠΙΝΑΚΑ ΠΟΥ ΒΡΙΣΚΕΤΑΙ ΣΤΗΝ 6^Η ΓΡΑΜΜΗ, ΕΝΩ Ο ΠΙΝΑΚΑΣ ΕΧΕΙ ΜΟΝΟ ΤΕΣΣΕΡΙΣ ΓΡΑΜΜΕΣ.

Γ. ΛΑΝΘΑΣΜΕΝΗ, ΓΙΑΤΙ ΑΡΙΣΤΕΡΑ ΑΠΟ ΤΗΝ ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ ΥΠΑΡΧΕΙ ΕΚΦΡΑΣΗ.

Δ. ΛΑΝΘΑΣΜΕΝΗ, ΓΙΑΤΙ ΟΙ ΔΕΙΚΤΕΣ ΤΟΥ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ ΠΡΕΠΕΙ ΝΑ ΕΧΟΥΝ ΠΑΝΤΑ ΑΚΕΡΑΙΗ ΤΙΜΗ.

Ε. ΣΩΣΤΗ. ΕΜΦΑΝΙΖΕΙ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΣΤΟΙΧΕΙΟΥ $A[6 \text{ DIV } 3,4]$, ΔΗΛΑΔΗ ΤΟ $A[2,4]$.

ΣΤ. ΛΑΝΘΑΣΜΕΝΗ, ΓΙΑΤΙ Ο ΠΙΝΑΚΑΣ Α ΕΙΝΑΙ ΔΙΣΔΙΑΣΤΑΤΟΣ ΚΑΙ ΜΕΣΑ ΣΤΙΣ ΑΓΚΥΛΕΣ ΠΡΕΠΕΙ ΝΑ ΥΠΑΡΧΟΥΝ ΔΥΟ ΔΕΙΚΤΕΣ.

25. Να γραφούν τμήματα προγραμμάτων που θα δημιουργούν τους παρακάτω πίνακες

9	8	8	8	8
7	9	8	8	8
7	7	9	8	8
7	7	7	9	8
7	7	7	7	9

-1	3	-1	5	-1
3	-1	5	-1	-1
-1	5	-1	-1	-1
5	-1	-1	-1	-1

A.
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
 ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
 ΑΝ I=J ΤΟΤΕ
 $A[I,J] \leftarrow 9$
 ΑΛΛΙΩΣ ΑΝ $I < J$ ΤΟΤΕ
 $A[I,J] \leftarrow 8$
 ΑΛΛΙΩΣ
 $A[I,J] \leftarrow 7$
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

B.
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4
 ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
 ΑΝ $I + J = 3$ ΤΟΤΕ
 $A[I,J] \leftarrow 3$
 ΑΛΛΙΩΣ ΑΝ $I + J = 5$ ΤΟΤΕ
 $A[I,J] \leftarrow 5$
 ΑΛΛΙΩΣ
 $A[I,J] \leftarrow -1$
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

26. Δίνεται τετραγωνικός πίνακας $A[N,N]$. Να γραφεί αλγόριθμος που θα εξετάζει αν είναι τριγωνικός κάτω εμφανίζοντας το μήνυμα «Είναι τριγωνικός κάτω» ή αν δεν είναι να εμφανίζει το μήνυμα «ο πίνακας δεν είναι τριγωνικός κάτω». Ένας πίνακας είναι τριγωνικός κάτω όταν όλα τα στοιχεία του που είναι πάνω από την κύρια διαγώνιο είναι μηδέν.

ΑΛΓΟΡΙΘΜΟΣ ΤΡΙΓΩΝΙΚΟΣ_ΚΑΤΩ
ΔΕΔΟΜΕΝΑ //N,A //

```

FLAG ← ΑΛΗΘΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Ν
        ΑΝ J > Ι ΤΟΤΕ
            ΑΝ A[I,J] <> 0 ΤΟΤΕ
                FLAG ← ΨΕΥΔΗΣ
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ FLAG = ΑΛΗΘΗΣ ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ 'Ο ΠΙΝΑΚΑΣ ΕΙΝΑΙ ..'
ΑΛΛΙΩΣ
    ΕΜΦΑΝΙΣΕ 'Ο ΠΙΝΑΚΑΣ ΔΕΝ ΕΙΝΑΙ ..'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ ΤΡΙΓΩΝΙΚΟΣ_ΚΑΤΩ

```

27. Να γραφεί πρόγραμμα που θα διαβάζει έναν πίνακα 50x100 αριθμών και στη συνέχεια θα υπολογίζει και εμφανίζει: α) το άθροισμα των στοιχείων του , β) το πλήθος των μη μηδενικών στοιχείων και γ) το ποσοστό των μηδενικών του στοιχείων.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Ι, J, ΠΛ, ΠΛ_Μ
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[50,100], ΑΘΡ, ΠΟΣ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 100
        ΔΙΑΒΑΣΕ Π[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΘΡ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 100
        ΑΘΡ← ΑΘΡ + Π[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΑΘΡ
ΠΛ ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 100
        ΑΝ Π [I,J] <>0 ΤΟΤΕ
            ΠΛ← ΠΛ + 1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ
ΠΛ_Μ←50*100 - ΠΛ
ΠΟΣ ← ΠΛ_Μ* 100/ (50*100)
ΓΡΑΨΕ ΠΟΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

28. Να γραφεί πρόγραμμα που θα διαβάζει έναν πίνακα 10x20 πραγματικών αριθμών. Να εμφανιστεί η μικρότερη τιμή του πίνακα και η θέση της στον πίνακα. Επίσης να εμφανιστεί η μεγαλύτερη αρνητική τιμή του πίνακα. Αν δεν υπάρχει αρνητική τιμή να εμφανιστεί το μήνυμα «Ο πίνακας δεν περιέχει αρνητικούς αριθμούς».

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Ι, J, ΓΜΙΝ, ΣΜΙΝ
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[10,20], ΜΙΝ, Χ, ΜΑΧ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
        ΔΙΑΒΑΣΕ Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΙΝ←Π[1,1]
ΓΜΙΝ←1
ΣΜΙΝ←1
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
        ΑΝ Π[Ι,J]<ΜΙΝ ΤΟΤΕ
            ΜΙΝ←Π[Ι,J]
            ΓΜΙΝ←Ι
            ΣΜΙΝ←J
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΙΝ, ΓΜΙΝ, ΣΜΙΝ
Χ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
        ΑΝ Π[Ι,J]<0 ΤΟΤΕ
            Χ←Π[Ι,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ Χ<0 ΤΟΤΕ
    ΜΑΧ←Χ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
        ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
            ΑΝ Π[Ι,J]>ΜΑΧ ΤΟΤΕ
                ΜΑΧ←Π[Ι,J]
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ ΜΑΧ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΔΕΝ ΥΠΑΡΧΟΥΝ ΑΡΝΗΤΙΚΟΙ ΑΡΙΘΜΟΙ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

29. Να γράφει πρόγραμμα που θα διαβάζει έναν πίνακα 10x20 και θα υπολογίζει και τυπώνει το άθροισμα κάθε γραμμής του πίνακα.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Ι, J
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[10,20], ΑΘΡ[10]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
        ΔΙΑΒΑΣΕ Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΑΘΡ[Ι]←0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
        ΑΘΡ[Ι]← ΑΘΡ[Ι]+Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΡΑΨΕ ΑΘΡ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

30. Να γραφεί πρόγραμμα που θα διαβάζει έναν πίνακα 10x20 και θα υπολογίζει και τυπώνει το άθροισμα κάθε στήλης του πίνακα.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Ι, J
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[10,20], ΑΘΡ[20]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
        ΔΙΑΒΑΣΕ Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
    ΑΘΡ[J]←0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
        ΑΘΡ[J]← ΑΘΡ[J]+Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 20
    ΓΡΑΨΕ ΑΘΡ[J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

31. Να γραφεί πρόγραμμα που θα διαβάζει έναν πίνακα 100x200 και θα υπολογίζει και τυπώνει: α) το μέγιστο κάθε γραμμής του, β) το ελάχιστο κάθε στήλης του, γ) το ποσοστό των μηδενικών στοιχείων ανά γραμμή και δ) το μέγιστο της 3^{ης} γραμμής του.

```

ΠΡΟΓΡΑΜΜΑ ΑΣΚ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Ι, J
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[100,200], ΜΑΧ[100], ΜΙΝ[200], ΠΛΗΘ[100], ΠΟΣ[100], ΜΑΧ1
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 200
        ΔΙΑΒΑΣΕ Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΜΑΧ[Ι]←Π[Ι,1]
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 200
        ΑΝ Π[Ι,J]>ΜΑΧ[Ι] ΤΟΤΕ
            ΜΑΧ[Ι]←Π[Ι,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΡΑΨΕ ΜΑΧ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 200
    ΜΙΝ[J]←Π[1,J]
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
        ΑΝ Π[Ι,J]<ΜΙΝ[J] ΤΟΤΕ
            ΜΙΝ[J]←Π[Ι,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 200
    ΓΡΑΨΕ ΜΙΝ[J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΠΛΗΘ[Ι]←0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 200
        ΑΝ Π[Ι,J]=0 ΤΟΤΕ
            ΠΛΗΘ[Ι]←ΠΛΗΘ[Ι]+1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΠΟΣ[Ι]←ΠΛΗΘ[Ι]*100/200*100
    ΓΡΑΨΕ ΠΟΣ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ1←Π[3,1]
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 200
    ΑΝ Π[3,J]>ΜΑΧ1 ΤΟΤΕ
        ΜΑΧ1←Π[3,J]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ1
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΣΚ

```

32. Να γραφεί πρόγραμμα που θα διαβάζει σε πίνακα 100x12 το μισθό 100 υπαλλήλων για καθένα από τους 12 μήνες του έτους. Να υπολογιστεί και εμφανιστεί: α) ο μέσος μηνιαίος μισθός ανά υπάλληλο, β) το σύνολο των μισθών για καθένα από τους μονούς μήνες, γ) το μέγιστο μισθό το μήνα Δεκέμβριο, δ) το σύνολο των αποδοχών των 30 πρώτων υπαλλήλων, ε) το σύνολο των μηνών όπου υπήρχε μισθός μικρότερος από 300 €.

```

ΠΡΟΓΡΑΜΜΑ ΜΙΣΘΟΙ
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: Μ[100,12], ΑΘΡ[100], ΜΟ[100], ΑΜ[12], ΜΑΧ, Α
    ΑΚΕΡΑΙΕΣ: Ι, J, ΠΛ[12], ΠΛΗΘ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
        ΔΙΑΒΑΣΕ Μ[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
ΑΘΡ[Ι]←0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
        ΑΘΡ[Ι]←ΑΘΡ[Ι] + Μ[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΜΟ[Ι]←ΑΘΡ[Ι]/12
ΓΡΑΨΕ ΜΟ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
ΑΜ[J]←0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
        ΑΝ J MOD 2<>0 ΤΟΤΕ
            ΑΜ[J]←ΑΜ[J]+Μ[Ι,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΝ J MOD 2<>0 ΤΟΤΕ
        ΓΡΑΨΕ ΑΜ[J]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ←-1
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΑΝ Π[Ι,12]>ΜΑΧ ΤΟΤΕ
        ΜΑΧ←Π[Ι,12]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ
Α←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
        Α←Α+Μ[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Α
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΠΛ[J]←0
        ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
            ΑΝ Μ[Ι,J]<300 ΤΟΤΕ
                ΠΛ[J]←ΠΛ[J]+1
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛΗΘ←0
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΝ ΠΛ[J]>=1 ΤΟΤΕ
        ΠΛΗΘ←ΠΛΗΘ+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛΗΘ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΙΣΘΟΙ

```

33. Να γραφεί πρόγραμμα που θα διαβάζει για 30 μαθητές το βαθμό τους στα «μαθηματικά», «Φυσική», «Πληροφορική», «Έκθεση» σε έναν δυσδιάστατο πίνακα. Οι τίτλοι των μαθημάτων να εκχωρηθούν σε νέο μονοδιάστατο πίνακα. α) να εμφανιστεί ο αριθμός των μαθητών με μέσο όρο στα 4 μαθήματα μικρότερο από 10, β) να υπολογιστεί ο αριθμός των μαθητών ανά μάθημα με βαθμό μικρότερο από 10 και να εμφανιστεί το όνομα του μαθήματος γ) να εμφανιστεί ο μέσος όρος στο μάθημα της πληροφορικής.

```

ΠΡΟΓΡΑΜΜΑ ΜΑΘ
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: Β[30,4], Α[30], ΜΟ[30], ΑΘΡ, Μ
    ΧΑΡΑΚΤΗΡΕΣ: Τ[4]

```

```

ΑΚΕΡΑΙΕΣ:Ι, J, ΠΛ, ΠΛΗΘ[4]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 4
        ΔΙΑΒΑΣΕ Β[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Τ[1]←'ΜΑΘΗΜΑΤΙΚΑ'
Τ[2]←'ΦΥΣΙΚΗ'
Τ[3]←'ΠΛΗΡΟΦΟΡΙΚΗ'
Τ[4]←'ΕΚΘΕΣΗ'
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    Α[Ι]←0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 4
        Α[Ι]← Α[Ι] + Β[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΜΟ[Ι]←Α[Ι]/4
    ΑΝ ΜΟ[Ι]<10 ΤΟΤΕ
        ΠΛ←ΠΛ+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 4
    ΠΛΗΘ[J]←0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
        ΑΝ Β[Ι,J]<10 ΤΟΤΕ
            ΠΛΗΘ[J]← ΠΛΗΘ[J]+1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΡΑΨΕ Τ[J], ΠΛΗΘ[J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΘΡ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    ΑΘΡ←ΑΘΡ+Β[Ι,3]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Μ←ΑΘΡ/30
ΓΡΑΨΕ Μ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΑΘ

```

34. Η ένωση ξενοδόχων του ν.Πιερίας αποφάσισε να καταγράψει για 150 ξενοδοχεία της τα κέρδη ή τη ζημία που παρουσίασαν για κάθε ένα μήνα του έτους εκτός των μηνών Νοεμβρίου και Φεβρουαρίου διότι δεν λειτούργησαν. Να αναπτύξετε πρόγραμμα που: α. Θα διαβάσει την ονομασία καθενός ξενοδοχείου και την αποθηκεύει στον πίνακα ΟΝ. β. Για κάθε ένα ξενοδοχείο θα διαβάσει τα κέρδη ή τη ζημία που είχε για κάθε ένα από τους 12 μήνες και θα τα αποθηκεύει στον πίνακα Κ, τοποθετώντας την τιμή 0 για τον Νοέμβριο, Φεβρουάριο. Η ζημία δείχνεται διαβάζοντας αρνητικό αριθμό. γ. Να εμφανίζει: 1. τις ονομασίες των ξενοδοχείων που παρουσίασαν κατά τη διάρκεια του έτους σε κάποιον μήνα κέρδη πάνω από 100.000€. 2. τις ονομασίες των ξενοδοχείων που παρουσίασαν τη μικρότερη ζημία.

ΠΡΟΓΡΑΜΜΑ ΞΕΝ


```

ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: I, J
  ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[150]
  ΠΡΑΓΜΑΤΙΚΕΣ: ΚΙΝ[150,12], ΜΑΧ
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 150
  ΔΙΑΒΑΣΕ ΟΝ[I]
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΝ J=11 Η΄ J=2 ΤΟΤΕ
      Κ[I,J]←0
    ΑΛΛΙΩΣ
      ΔΙΑΒΑΣΕ Κ[I,J]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 150
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΝ J<>11 ΚΑΙ J<>2 ΤΟΤΕ
      ΑΝ Κ[I,J]>100000 ΤΟΤΕ
        ΓΡΑΨΕ ΟΝ[I]
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ<-1000000000000000000
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 150
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΝ Κ[I,J]<0 ΚΑΙ J<>11 ΚΑΙ J<>2 ΤΟΤΕ
      ΑΝ Κ[I,J]>ΜΑΧ ΤΟΤΕ
        ΜΑΧ←Κ[I,J]
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 150
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
    ΑΝ Κ[I,J]=ΜΑΧ ΚΑΙ J<>11 ΚΑΙ J<>2 ΤΟΤΕ
      ΓΡΑΨΕ ΟΝ[I]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΞΕΝ

```

35. Σε ένα νομό της χώρας, υπάρχουν 15 κέντρα μεταχειρισμένων αυτοκινήτων που το καθένα διαθέτει από 10 έως 20 διαφορετικά μοντέλα προς πώληση. Να γραφεί πρόγραμμα που: α. για κάθε κέντρο μεταχειρισμένων, θα διαβάζει την ονομασία του και τον αριθμό των μοντέλων και τα καταχωρίζει στους πίνακες ΟΝ[15], ΑΡ[15] πραγματοποιώντας έλεγχο εγκυρότητας ώστε ο αριθμός να είναι από 0 μέχρι και 20. β. για κάθε κέντρο θα διαβάζει για το κάθε μοντέλο που διαθέτει, την τιμή πώλησης και θα τη καταχωρίζει στον πίνακα Π. γ. να υπολογίζει και εμφανίζει: 1. από όλα τα κέντρα πόσα μοντέλα αυτοκινήτων υπάρχουν με τιμή πώλησης από 8000 έως και 12000€. 2. από όλα τα κέντρα την τιμή του πιο φτηνού μοντέλου και 3. το όνομα του κάθε κέντρου όπου υπάρχει το πιο φτηνό μοντέλο.

```

ΠΡΟΓΡΑΜΜΑ ΑΥΤ
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: I, J, ΑΡ[15], ΠΛΗΘ

```

```

ΧΑΡΑΚΤΗΡΕΣ:ΟΝ[15]
ΠΡΑΓΜΑΤΙΚΕΣ: Π[15,20], ΦΤΗΝΟ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 15
    ΔΙΑΒΑΣΕ ΟΝ[Ι]
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ ΑΡ[Ι]
        ΜΕΧΡΙΣ_ΟΤΟΥ ΑΡ[Ι]>=0 ΚΑΙ ΑΡ[Ι]<=20
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 15
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΑΡ[Ι]
        ΔΙΑΒΑΣΕ Π[Ι,J]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛΗΘ←0
ΦΤΗΝΟ←Π[1,1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 15
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΑΡ[Ι]
        ΑΝ Π[Ι,J]>=8000 ΚΑΙ Π[Ι,J]<=12000 ΤΟΤΕ
            ΠΛΗΘ←ΠΛΗΘ+1
            ΤΕΛΟΣ_ΑΝ
        ΑΝ Π[Ι,J]<ΦΤΗΝΟ ΤΟΤΕ
            ΦΤΗΝΟ←Π[Ι,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛΗΘ, ΦΤΗΝΟ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 15
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΑΡ[Ι]
        ΑΝ Π[Ι,J]= ΦΤΗΝΟ ΤΟΤΕ
            ΓΡΑΨΕ ΟΝ[Ι]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΥΤ

```

36. Να γραφεί πρόγραμμα που: α) θα διαβάζει το όνομα 89 φοιτητών (όνομα διάφορο του χαρακτήρα του κενού), β) θα διαβάζει τις ψήφους που πήραν σε καθεμία από τις 3 ψηφοφορίες για την ανάδειξη εκπροσώπων (οι ψήφοι να μην είναι αρνητικός αριθμός), γ) να διαβάζεται το όνομα ενός φοιτητή και αν συμμετείχε στην ψηφοφορία να εμφανίζεται «ΝΑΙ» και οι ψήφοι του σε κάθε ψηφοφορία αλλιώς να εμφανίζεται «ΟΧΙ», δ) με δεδομένο ότι εκλέγεται ο φοιτητής με το μεγαλύτερο σύνολο ψήφων να εμφανίζεται το όνομα του εκλεγέντος (θεωρούμε ότι είναι ένας).

```

ΠΡΟΓΡΑΜΜΑ ΨΗΦ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Ι, J,
    ΠΡΑΓΜΑΤΙΚΕΣ: Ψ[89,3], Α[89], ΜΑΧ
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[89], ΟΝΟΜΑ, ΟΜΑΧ
    ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 89
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ ΟΝ[Ι]
        ΜΕΧΡΙΣ_ΟΤΟΥ ΟΝ[Ι]<> ``
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 89
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 3

```

```

        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            ΔΙΑΒΑΣΕ Ψ[I,J]
            ΜΕΧΡΙΣ_ΟΤΟΥ Ψ[I,J]>=0
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ ΟΝΟΜΑ
    ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 89
        ΑΝ ΟΝ[I]=ΟΝΟΜΑ ΤΟΤΕ
            ΓΡΑΨΕ 'ΝΑΙ'
            ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 3
                ΓΡΑΨΕ Ψ[I,J]
                ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ
            ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΤΟΤΕ
        ΓΡΑΨΕ 'ΟΧΙ'
    ΤΕΛΟΣ_ΑΝ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 89
        Α[I]←0
            ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 3
                Α[I]←Α[I]+Ψ[I,J]
            ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΑΧ←-1
    ΟΜΑΧ←``
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 89
        ΑΝ Α[I]>ΜΑΧ ΤΟΤΕ
            ΜΑΧ←Α[I]
            ΟΜΑΧ←ΟΝ[I]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ ΜΑΧ, ΟΜΑΧ
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΨΗΦ

```

37. Η εταιρεία έτοιμου φαγητού «Γιώργος ΑΕ» έχει 20 εστιατόρια σε διάφορα νησιά. Να γραφεί πρόγραμμα που: α. Διαβάσει την ονομασία του κάθε ενός εστιατορίου και τις καταχωρίζει στον πίνακα ΟΝ. β. Να διαβάσει τις μηνιαίες εισπράξεις του κάθε εστιατορίου για το έτος 2019 και τις καταχωρίζει στον πίνακα ΕΙΣ. γ. Να εμφανίζει: 1. τα ονόματα των εστιατορίων με ετήσιο σύνολο εισπράξεων πάνω από 50000€. 2. το πλήθος των εστιατορίων που είχαν μικρότερο σύνολο εισπράξεων στο α' εξαμ. από ότι στο β'. 3. το όνομα ή τα ονόματα των εστιατορίων με τις λιγότερες εισπράξεις για το μήνα Ιούλιο.

```

ΠΡΟΓΡΑΜΜΑ ΦΑΓ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, J, ΠΛΗΘΟΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΕΙΣ[20,12], ΑΘΡ, ΑΘΡ1, ΑΘΡ2, ΜΙΝ
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[20]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    ΔΙΑΒΑΣΕ ΟΝ[I]
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12
        ΔΙΑΒΑΣΕ ΕΙΣ[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    ΑΘΡ←0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12

```

```

        ΑΘΡ←ΑΘΡ+ΕΙΣ[Ι,Ι]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ ΑΘΡ>50000 ΤΟΤΕ
        ΓΡΑΨΕ ΟΝ[Ι]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛΗΘΟΣ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    ΑΘΡ1←0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
        ΑΘΡ1←ΑΘΡ1+ΕΙΣ[Ι,Ι]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΘΡ2←0
    ΓΙΑ Ι ΑΠΟ 7 ΜΕΧΡΙ 12
        ΑΘΡ2←ΑΘΡ2+ΕΙΣ[Ι,Ι]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ ΑΘΡ1<ΑΘΡ2 ΤΟΤΕ
        ΠΛΗΘΟΣ←ΠΛΗΘΟΣ+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛΗΘΟΣ
ΜΙΝ←ΕΙΣ[1,7]
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 20
    ΑΝ ΕΙΣ[Ι,7]<ΜΙΝ ΤΟΤΕ
        ΜΙΝ←ΕΙΣ[Ι,7]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    ΑΝ ΕΙΣ[Ι,7] = ΜΙΝ ΤΟΤΕ
        ΓΡΑΨΕ ΟΝ[Ι]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΦΑΓ

```

38. Ένα κατάστημα ηλεκτρονικών ειδών θέλει να καταχωρίσει σε πίνακες τις ονομασίες 4 μοντέλων ραδιοφώνων και τον αριθμό τεμαχίων που πούλησε καθένας από τους 10 πωλητές του. Επίσης θέλει να αποθηκεύσει σε πίνακα τα ονόματα των 10 πωλητών. Αφού διαβαστούν τα δεδομένα με τη βοήθεια προγράμματος να υπολογιστεί και εμφανιστεί: α) το όνομα του δεύτερου μοντέλου και ο αριθμός τεμαχίων που πούλησε το κατάστημα από το συγκεκριμένο μοντέλο, β) το όνομα και ο μέσος όρος πωλήσεων του 5^{ου} πωλητή, γ) ο συνολικός μέσος όρος πωλήσεων από τους 3 τελευταίους πωλητές, δ) τα ονόματα των πωλητών ταξινομημένα σε φθίνουσα σειρά με βάση το μέσο όρο πωλήσεων τους μετά το μήνυμα «Φθίνουσα σειρά πωλήσεων».

```

ΠΡΟΓΡΑΜΜΑ ΠΩΛ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, Ι,
    ΠΡΑΓΜΑΤΙΚΕΣ: Π[4,10], Α, ΑΘ[4], ΜΟ, ΑΘΡ, ΜΟ1, Α[10], TEMP1, TEMP2
    ΧΑΡΑΚΤΗΡΕΣ: Μ[4], ΟΝ[10]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΔΙΑΒΑΣΕ Μ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
        ΔΙΑΒΑΣΕ Π[Ι,Ι]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
    ΔΙΑΒΑΣΕ ΟΝ[J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Α←0
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
    Α←Α+Π[2,J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Μ[2], Α
ΑΘ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΑΘ←ΑΘ+Π[Ι,5]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ←ΑΘ/4
ΓΡΑΨΕ ΟΝ[5], ΜΟ
ΑΘΡ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΙΑ J ΑΠΟ 8 ΜΕΧΡΙ 10
        ΑΘΡ←ΑΘΡ+Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ1←ΑΘΡ/12
ΓΡΑΨΕ ΜΟ1
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
Α[J]←0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 4
        Α[J]←Α[J]+ Π[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ[J]←Α[J]/4
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 10 ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
        ΑΝ ΜΟ[J-1]<ΜΟ[J] ΤΟΤΕ
            ΤΕΜΡ1 ← ΜΟ[J-1]
            ΜΟ[J-1] ← ΜΟ[J]
            ΜΟ[J] ← ΤΕΜΡ1
            ΤΕΜΡ2←ΟΝ[J-1]
            ΟΝ[J-1]←ΟΝ[J]
            ΟΝ[J]←ΤΕΜΡ2
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ `ΦΘΙΝΟΥΣΑ ΣΕΙΡΑ ΠΩΛΗΣΕΩΝ`
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΡΑΨΕ ΟΝ[J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΠΩΛ

```

39. Στο πρωτάθλημα ποδοσφαίρου συμμετέχουν 18 ομάδες στη σούπερ λιγκ, αγωνίζονται 360 ποδοσφαιριστές και η αγωνιστική περίοδος ολοκληρώνεται σε 24 αγωνιστικές. Η διοργανώτρια αρχή καταγράφει σε κατάλληλους πίνακες τα τέρματα που πέτυχε ο παίκτης κάθε ομάδας σε κάθε μια αγωνιστική. Να αναπτύξετε πρόγραμμα που:

α. Για κάθε ένα από τους 360 παίκτες: 1. διαβάζει το ονοματεπώνυμο και το καταχωρίζει στο πίνακα Π, την ομάδα που αγωνίζεται και τη καταχωρίζει στο πίνακα Ο. 2. τα τέρματα που πέτυχε σε κάθε μία από τις 24 αγωνιστικές και τα καταχωρίζει στον πίνακα ΓΚΟΛ. β. Να βρίσκει και να εμφανίζει: 1. για κάθε μια αγωνιστική τα ονοματεπώνυμα των παικτών που σημείωσαν τα περισσότερα τέρματα, καθώς και σε ποια ομάδα αγωνίζονται. 2. τις αγωνιστικές κατά τις οποίες σημειώθηκαν τα περισσότερα συνολικά τέρματα. 3. τα ονοματεπώνυμα των παικτών που δεν σημείωσαν κανένα τέρμα σε ολόκληρο το

πρωτάθλημα. 4. το όνομα του πρώτου σκόρερ του πρωταθλήματος και την ομάδα που αγωνίζεται.

```
ΠΡΟΓΡΑΜΜΑ ΠΡΩΤ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, J, ΘΕΣΗ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΓΚΟΛ[360,24], ΜΑΧ, ΑΘΡ[24], ΣΥΝ[360]
    ΧΑΡΑΚΤΗΡΕΣ: Π[360], Ο[360]
    ΛΟΓΙΚΕΣ: FLAG
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 360
    ΔΙΑΒΑΣΕ Π[I], Ο[I]
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 24
        ΔΙΑΒΑΣΕ ΓΚΟΛ[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 24
    ΜΑΧ←ΓΚΟΛ[1,J]
    ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 360
        ΑΝ ΓΚΟΛ[I,J]>ΜΑΧ ΤΟΤΕ
            ΜΑΧ←ΓΚΟΛ[I,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 360
        ΑΝ ΓΚΟΛ[I,J]=ΜΑΧ ΤΟΤΕ
            ΓΡΑΨΕ ΟΝ[I], Π[I]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 24
    ΑΘΡ[J]←0
    ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 360
        ΑΘΡ[J]←ΑΘΡ[J]+ΓΚΟΛ[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ←ΑΘΡ[1]
ΓΙΑ J ΑΠΟ 2 ΜΕΧΡΙ 24
    ΑΝ ΑΘΡ[J]>ΜΑΧ ΤΟΤΕ
        ΜΑΧ←ΑΘΡ[J]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 24
    ΑΝ ΑΘΡ[J]= ΜΑΧ ΤΟΤΕ
        ΓΡΑΨΕ J
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 360
    FLAG←ΑΛΗΘΗΣ
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 24
        ΑΝ ΓΚΟΛ [I,J]<>0 ΤΟΤΕ
            FLAG←ΨΕΥΔΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ FLAG= ΑΛΗΘΗΣ ΤΟΤΕ
        ΓΡΑΨΕ Π[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 360
    ΣΥΝ[I]←0
```

```

    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 24
        ΣΥΝ[I]←ΣΥΝ[I]+ΓΚΟΛ[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ←ΣΥΝ[1]
ΘΕΣΗ←1
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 360
    ΑΝ ΣΥΝ[I]>ΜΑΧ ΤΟΤΕ
        ΜΑΧ←ΣΥΝ[I]
        ΘΕΣΗ←I
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Ο[ΘΕΣΗ], Π[ΘΕΣΗ]
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΠΡΩΤ

```

40. Η βιβλιοθήκη του Δήμου Δ έχει καταχωρίσει 250 τίτλους βιβλίων στον πίνακα T[250] και αποθηκεύει στους πίνακες Δ1[250,6] και Δ2[250,6] τον αριθμό των ατόμων που δανείστηκαν κάθε ένα βιβλίο ανά μήνα για το 1^ο και το 2^ο εξάμηνο του έτους. Επίσης στον πίνακα Μ[12] υπάρχουν καταχωρισμένα τα ονόματα των μηνών του έτους με τη γνωστή σειρά. Να γραφεί πρόγραμμα που: α. ενώνει τους πίνακες Δ1 και Δ2 σε ένα πίνακα Δ[250,12] ο οποίος θα περιέχει τον δανεισμό του έτους. β. Διαβάζει τον τίτλο ενός βιβλίου και εκτυπώνει το όνομα του μήνα κατά τον οποίο τα περισσότερα άτομα το δανείστηκαν. Αν δεν υπάρχει το βιβλίο θα εκτυπώνει κατάλληλο μήνυμα. γ. Εκτυπώνει τους τίτλους των βιβλίων και τον αριθμό των ατόμων που δανείστηκαν το κάθε βιβλίο το μήνα Μάρτιο, ταξινομημένα κατά φθίνουσα σειρά.

```

ΠΡΟΓΡΑΜΜΑ ΒΙΒΛ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I,J, Δ1[250,6], Δ2[250,6], Δ[250,12], ΒΠ[250], TEMP1
    ΧΑΡΑΚΤΗΡΕΣ: Τ[250], ΤΙΤΛΟΣ, Μ[12], ΒΤ[25], TEMP2
    ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 250
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 6
        Δ[I,J]←Δ1[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 250
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 6
        Δ[I,J+6]←Δ2[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ ΤΙΤΛΟΣ
I←1
ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ
ΟΣΟ I<=250 ΚΑΙ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ Τ[I]=ΤΙΤΛΟΣ ΤΟΤΕ
        ΜΑΧ←Δ[I,1]
        ΘΕΣΗ←1
        ΓΙΑ J ΑΠΟ 2 ΜΕΧΡΙ 12
            ΑΝ Δ[I,J]>ΜΑΧ ΤΟΤΕ
                ΜΑΧ←Δ[I,J]
                ΘΕΣΗ←J
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΡΑΨΕ Μ[ΘΕΣΗ]
        ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ

```

```

        I←I+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Ο ΤΙΤΛΟΣ ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 250
    ΒΠ[I]←Δ[I,3]
    ΒΤ[I]←Τ[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 250
    ΓΙΑ J ΑΠΟ 250 ΜΕΧΡΙ I ΜΕ_ΒΗΜΑ -1
        ΑΝ ΒΠ[J-1]<ΒΠ[J] ΤΟΤΕ
            TEMP1←ΒΠ[J-1]
            ΒΠ[J-1]←ΒΠ[J]
            ΒΠ[J]←TEMP1
            TEMP2←ΒΤ[J-1]
            ΒΤ[J-1]←ΒΤ[J]
            ΒΤ[J]←TEMP2
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 250
    ΓΡΑΨΕ ΒΠ[I], ΒΤ[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΒΙΒΛ

```

Ασκήσεις

21. Η ΕΜΥ μελετάει τις θερμοκρασίες σε διάφορες πόλεις της Ελλάδας. Καταχωρούνται λοιπόν τα ονόματα των 100 πόλεων που συμμετέχουν στην έρευνα καθώς και οι θερμοκρασίες των πόλεων αυτών τον μήνα που πέρασε. Να αναπτυχθεί πρόγραμμα όπου:

- Θα διαβάζει τα απαραίτητα στοιχεία. Να περιγραφούν οι δομές δεδομένων που θα χρησιμοποιηθούν
- Θα εντοπίζει και θα εκτυπώνει το όνομα της κατά μέσο όρο θερμότερης πόλης του μήνα
- Θα εντοπίζει για κάθε πόλη τις μέρες του μήνα όπου υπάρχει θερμοκρασία μεγαλύτερη από την προηγούμενη και την επόμενη μέρα.

22. Να αναπτυχθεί πρόγραμμα που θα επεξεργάζεται τα στοιχεία των αγώνων ποδοσφαίρου. Θα καταχωρεί σε δισδιάστατο πίνακα τα γκολ που πέτυχε κάθε μια από τις 16 ομάδες του πρωταθλήματος σε κάθε μία από τις 32 αγωνιστικές του πρωταθλήματος και θα εμφανίζει:

- Ποιά ομάδα και σε ποια αγωνιστική πέτυχε τα περισσότερα γκολ;
- Την καλύτερη επίθεση του πρωταθλήματος
- Την χειρότερη επίθεση του πρωταθλήματος
- Σε πόσους αγώνες κάθε ομάδα πέτυχε περισσότερα γκολ από το μέσο όρο της;

23. Η γραμματεία του Λυκείου καταχωρεί τα ονόματα των 100 μαθητών της Α' Λυκείου και τους 10 βαθμούς του κάθε μαθητή στα μαθήματα που βαθμολογήθηκε. Να αναπτύξετε πρόγραμμα που:

- θα διαβάζει τα παραπάνω στοιχεία σε κατάλληλους πίνακες.
- θα εκτυπώνει το όνομα του 5ου μαθητή και πόσες φορές έχει βαθμό άριστα (άριστα είναι ≥ 18).
- θα εκτυπώνει το μέσο όρο του 23ου μαθητή.
- θα εκτυπώνει το μεγαλύτερο βαθμό στο 3ο μάθημα και το όνομα του μαθητή.
- θα εκτυπώνει το μέσο όρο στο 1ο μάθημα.

24. Σε πίνακες Π2005[200,12] και Π2006[200,12] έχουμε αποθηκεύσει τις μηνιαίες πωλήσεις των 200 σημείων πώλησης της εταιρείας για τα έτη 2005 και 2006 αντίστοιχα. Να αναπτύξετε πρόγραμμα που αφού διαβάσει τους πίνακες θα εμφανίζει:

1. Ποιο σημείο πώλησης και σε ποιο μήνα επέτυχε τις περισσότερες πωλήσεις το 2005 και ποιο το 2006; Πρόκειται για το ίδιο σημείο πώλησης;
2. Ο Μάιος του 2005 ήταν πιο προσοδοφόρος ή αυτός του 2006;
3. Πόσες φορές το σημείο πώλησης 33 είχε μεγαλύτερες πωλήσεις το 2005 και πόσες λιγότερες σε σχέση με το 2006.
4. Το 2005 ήταν καλύτερο για την εταιρεία ή το 2006;

25. Διαβάζουμε σε κατάλληλο πίνακα τις μηνιαίες πωλήσεις των 200 πωλητών της εταιρείας μας. Να αναπτύξετε πρόγραμμα που:

1. θα εμφανίζει πόσοι πωλητές έχουν επιτύχει συνολικό ποσό πωλήσεων μεγαλύτερο από 1000€ κατά τους μήνες Ιούνιο-Ιούλιο-Αύγουστο.
2. Πόσοι πωλητές είχαν το Σεπτέμβριο περισσότερες πωλήσεις από τον Μάιο.
3. Πόσοι πωλητές ήταν καλύτεροι το 2ο τρίμηνο σε σχέση με το 1ο.
4. Ο 10ος πωλητής ποιους μήνες επέτυχε πωλήσεις περισσότερες από 2000€;
5. Ο 100ος πωλητής με έναρξη τον Ιανουάριο ποιο μήνα επέτυχε συγκεντρωτικές πωλήσεις περισσότερες από 10.000€; Αν δεν ξεπέρασε το όριο αυτό να εμφανίζεται κατάλληλο μήνυμα.

26. Καταχωρούμε σε πίνακα ON τα ονόματα των 130 μαθητών της Γ' Γυμνασίου του δήμου Αλυζίας και σε πίνακα B τους βαθμούς κάθε μαθητή σε κάθε ένα από τα 10 μαθήματα. Να αναπτύξετε πρόγραμμα που αφού διαβάζει τα στοιχεία των πινάκων ON και B:

- a. να εμφανίζει ποιος μαθητής (μοναδικός είναι) έχει το μεγαλύτερο μέσο όρο, που να είναι όμως μικρότερος από 17.
- β. να εμφανίζει για κάθε μαθητή, ποιος είναι ο μέσος όρος των βαθμών του που βρίσκονται στο διάστημα [14,18].
- γ. να εμφανίζει τους μαθητές που είχαν τις περισσότερες φορές τον υψηλότερο βαθμό σε κάποιο μάθημα.
- δ. να εμφανίζει τα ονόματα των μαθητών που είχαν τουλάχιστον δυο βαθμούς άριστα (≥ 18).

27. Μια εταιρεία διανομής κινηματογραφικών ταινιών έχει καταγράψει στους παρακάτω πίνακες τα στοιχεία για την διακίνηση των ταινιών της παγκόσμια, για τη χρονιά που πέρασε: πίνακας ΤΑΙΝΙΑ[200] που περιέχει τους τίτλους των ταινιών, πίνακας ΧΩΡΑ[120] που περιέχει τα ονόματα των 120 χωρών στις οποίες η εταιρεία δραστηριοποιείται και πίνακα ΕΙΣΠΡΑΞΕΙΣ[200,120] που περιέχει τις εισπράξεις από την προβολή κάθε ταινίας σε κάθε χώρα (αν κάποια ταινία δεν προβλήθηκε σε κάποια χώρα υπάρχει η τιμή μηδέν). Να αναπτύξετε πρόγραμμα που θα διαβάζει τους παραπάνω πίνακες και στη συνέχεια:

1. θα εκτυπώνει τις ταινίες που έχουν τουλάχιστον σε 3 χώρες εισπράξεις μεγαλύτερες από 100.000€. Θα εκτυπωθεί μια φορά το όνομα κάθε ταινίας.
2. θα διαβάζει το όνομα μιας χώρας και θα εκτυπώνει πόσες ταινίες έχουν προβληθεί σε αυτή.
3. θα διαβάζει τον τίτλο μιας ταινίας και θα εκτυπώνει τις χώρες που προβλήθηκε με διάταξη από αυτήν με τις περισσότερες εισπράξεις προς αυτήν με τις λιγότερες. Οι χώρες που η ταινία δεν προβλήθηκε να μην εκτυπώνονται.

28.

Στις κατατακτήριες εξετάσεις που διοργάνωσε το τμήμα Αυτοματισμού, διαγωνίστηκαν 80 υποψήφιοι φοιτητές σε τέσσερα μαθήματα: Ηλεκτρολογία, Μαθηματικά, Φυσική, Ηλεκτρονικά και βαθμολογήθηκαν στο κάθε ένα μάθημα με έναν ακέραιο βαθμό από το 1 έως και το 10. Να κατασκευάσετε πρόγραμμα σε ΓΛΩΣΣΑ που:

- α. Θα καταχωρίζει τα τέσσερα μαθήματα με τη σειρά με την οποία δίνονται στον πίνακα ΜΑΘ.
- β. Θα διαβάσει τα ονόματα των διαγωνιζομένων και τους βαθμούς που έλαβαν στα τέσσερα μαθήματα, και θα τα καταχωρίζει σε κατάλληλους πίνακες.
- γ. Για κάθε μάθημα θα υπολογίζει τη μέση επίδοση των υποψηφίων και να εκτυπώνει τα ονόματα των υποψηφίων που έγραψαν πάνω από τη μέση επίδοση.
- δ. Να εκτυπώνει τα ονόματα όσων υποψηφίων έγραψαν κάτω από τη μέση επίδοση στα μαθήματα της Ηλεκτρολογίας και της Φυσικής μαζί.
- ε. Να εκτυπώνει το μάθημα που είχε τους περισσότερους υποψηφίους με τη χαμηλότερη βαθμολογία.

29.

Μία δισκογραφική εταιρεία καταγράφει στοιχεία για ένα έτος για κάθε ένα από τα 20 CDs που κυκλοφόρησε. Τα στοιχεία αυτά είναι ο τίτλος του CD, ο τύπος της μουσικής που περιέχει και οι μηνιαίες του πωλήσεις (ποσά σε ευρώ) στη διάρκεια του έτους. Οι τύποι μουσικής είναι δύο: «ορχηστρική» και «φωνητική». Να αναπτυχθεί αλγόριθμος ο οποίος:

- α. Για κάθε ένα από τα 20 CDs, να διαβάσει τον τίτλο, τον τύπο της μουσικής και τις πωλήσεις του για κάθε μήνα, ελέγχοντας την έγκυρη καταχώριση του τύπου της μουσικής.
- β. Να εμφανίζει τον τίτλο ή τους τίτλους των CDs με τις περισσότερες πωλήσεις τον 3^ο μήνα του έτους.
- γ. Να εμφανίζει τους τίτλους των ορχηστρικών CDs με ετήσιο σύνολο πωλήσεων τουλάχιστον 5000 ευρώ.
- δ. Να εμφανίζει πόσα από τα CDs είχαν σύνολο πωλήσεων στο δεύτερο εξάμηνο μεγαλύτερο απ' ό,τι στο πρώτο. (Ημερήσια 2007)

30.

Η γραμματεία ενός ΙΕΚ καταγράφει για κάθε έναν από τους 200 σπουδαστές τα εξής στοιχεία: το ονοματεπώνυμο, την ειδικότητα που σπουδάζει και τις εβδομαδιαίες απουσίες του για τις 14 εβδομάδες των μαθημάτων ενός εξαμήνου. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που:

- α. Για κάθε σπουδαστή θα διαβάσει:
 1. το ονοματεπώνυμο και την ειδικότητα που σπουδάζει και να τα καταχωρίζει σε μονοδιάστατους πίνακες,
 2. τις απουσίες του για κάθε μια εβδομάδα του εξαμήνου και να τις καταχωρίζει σε δισδιάστατο πίνακα.
- β. Θα εμφανίζει το όνομα και την ειδικότητα του σπουδαστή που έχει τις λιγότερες απουσίες την 10η εβδομάδα.
- γ. Θα διαβάσει το όνομα ενός σπουδαστή και θα εμφανίζει τις απουσίες του ταξινομημένες κατά φθίνουσα σειρά για όλες τις εβδομάδες του εξαμήνου. Αν ο σπουδαστής δεν βρεθεί, να εμφανίζει κατάλληλο μήνυμα.

31.

Ένας εργολάβος έχει στην κατοχή του μία πολυκατοικία 12 ορόφων με 8 διαμερίσματα ανά όροφο, τα οποία και έχει ενοικιάσει με μηνιαίο μίσθωμα διαμερίσματος από 300 € μέχρι και 550 €. Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

- α. Θα διαβάζει για κάθε όροφο το μίσθωμα του τρέχοντος μήνα κάθε διαμερίσματος και θα το καταχωρίζει στον πίνακα EN[12, 8], πραγματοποιώντας έλεγχο ορθής καταχώρισης του μισθώματος.
- β. Θα εμφανίζει τα τρία υψηλότερα μηνιαία μισθώματα έναντι των οποίων έχουν ενοικιαστεί κάποια από τα διαμερίσματα του 4ου ορόφου της πολυκατοικίας.

Παρατήρηση: Το μηνιαίο μίσθωμα είναι ένας ακέραιος αριθμός.

32. Η εταιρεία DeltaTime Systems εξοπλίζει αθλητικούς αγώνες με συστήματα χρονομέτρησης. Τα συστήματα αυτά δημιουργούν δυο παράλληλους πίνακες, τον πίνακα ΚΩΔΙΚΟΣ με τον κωδικό αριθμό κάθε αθλητή και τον πίνακα ΧΡΟΝΟΣ με τον χρόνο που χρειάστηκε ο αθλητής να τερματίσει. (Οι πίνακες είναι ταξινομημένοι ως προς τον χρόνο). Η διοργανώτρια επιτροπή διαθέτει τον πίνακα ΑΘΛΗΤΗΣ που περιέχει ως πρώτη στήλη το όνομα κάθε αθλητή και ως δεύτερη στήλη τον αντίστοιχο κωδικό του. Να αναπτύξετε πρόγραμμα που, με δεδομένους τους παραπάνω πίνακες:

- α. Θα δημιουργεί τον πίνακα ΘΕΣΗ που είναι παράλληλος με τον πίνακα ΑΘΛΗΤΗΣ και περιέχει τη θέση που κατέλαβε κάθε αθλητής.
- β. Θα εκτυπώνει αλφαβητικά τα ονόματα των αθλητών και τη θέση που κατέλαβαν.

33.

Στο τελευταίο φεστιβάλ ψηφιακής δημιουργίας συμμετείχαν 10 ομάδες μαθητών. Κάθε ομάδα παρουσίασε μια εργασία. Από κάθε ομάδα ζητήθηκε να βαθμολογήσει όλες τις εργασίες, τόσο τη δική της όσο και των υπολοίπων 9 ομάδων. Να κατασκευάσετε πρόγραμμα το οποίο:

- α. Να περιλαμβάνει κατάλληλο τμήμα δηλώσεων.
- β. Να καταχωρίζει:
 1. τα ονόματα των ομάδων, σε πίνακα O[10].
 2. τους ακέραιους βαθμούς, σε πίνακα B[10,10]. Οι βαθμοί να εισάγονται, για κάθε ομάδα με τη σειρά, από την πρώτη μέχρι τη δέκατη, ως εξής:
 - να εισάγεται πρώτα ο βαθμός που έδωσε στη δική της εργασία.
 - για καθεμιά από τις υπόλοιπες ομάδες, με τη σειρά, που έχουν καταχωριστεί στον πίνακα O, να εμφανίζεται το όνομά της και να εισάγεται ο αντίστοιχος βαθμός.
- γ. Να εμφανίζει το όνομα της ομάδας που συγκέντρωσε τον μεγαλύτερο μέσο όρο βαθμολογίας. Κατά τον υπολογισμό του μέσου όρου να εξαιρούνται ο μεγαλύτερος και ο μικρότερος βαθμός της.
- δ. Να εμφανίζει το όνομα της ομάδας η οποία βαθμολόγησε τον εαυτό της πλησιέστερα στον μέσο όρο των βαθμών που έλαβε από τις υπόλοιπες ομάδες.

Για το ερώτημα γ να θεωρήσετε ότι οι τιμές του μέσου όρου, του μικρότερου και του μεγαλύτερου βαθμού είναι μοναδικές. Για το ερώτημα δ να θεωρήσετε ότι η τιμή του μέσου όρου είναι μοναδική.

34. Ο Ελληνικός Οργανισμός Τουρισμού αποφάσισε να διεξαγάγει μια έρευνα για τα ελληνικά ξενοδοχεία. Στην έρευνα συμμετέχουν τα 7000 ξενοδοχεία της επικράτειας, και γι' αυτά καταγράφονται οι μηνιαίες εισπράξεις για το περασμένο έτος. Να αναπτυχθεί πρόγραμμα που:

- α. Θα διαβάζει τα απαραίτητα στοιχεία - πίνακες.

- β. Θα υπολογίζει τις ετήσιες εισπράξεις κάθε ξενοδοχείου και θα τις εκχωρεί σε έναν νέο πίνακα.**
- γ. Θα εκτυπώνει το όνομα του ξενοδοχείου με τις περισσότερες εισπράξεις (είναι μοναδικό).**
- δ. Θα υπολογίζει και θα εκτυπώνει το όνομα κάθε ξενοδοχείου συνοδευόμενο από το πλήθος των μηνών που λειτούργησε (δηλαδή είχε έσοδα) την περασμένη χρονιά.**
- ε. Θα δημιουργεί και θα εκτυπώνει τον πίνακα ΤΡΙΜΗΝΑ[4], που περιέχει τις εισπράξεις που παρουσιάστηκαν ανά τρίμηνο.**

35.

Μία εταιρεία παράγει χυμούς σε δύο μεγάλα εργοστάσια, το Α και το Β. Για την παραγωγή των χυμών χρησιμοποιούνται τέσσερις πρώτες ύλες: νερό και τα φρούτα πορτοκάλι, ροδάκινο, μήλο. Η κεντρική διαχείριση της εταιρείας λαμβάνει στο τέλος του χρόνου από τον διευθυντή του κάθε εργοστασίου έναν πίνακα με τις ποσότητες σε πρώτη ύλη που χρησιμοποιήθηκαν σε αυτό για την παραγωγή χυμών. Να κατασκευάσετε πρόγραμμα σε ΓΛΩΣΣΑ που θα υλοποιεί τον παρακάτω αλγόριθμο:

- α. Θα τοποθετεί στον μονοδιάστατο πίνακα ON τις ονομασίες της κάθε πρώτης ύλης με τη σειρά με την οποία δίνονται.
- β. Θα διαβάζει τις ποσότητες σε κιλά που χρησιμοποιήθηκαν από το κάθε ένα υλικό για κάθε ένα μήνα του έτους 2009 στο εργοστάσιο Α, και να τα τοποθετεί στον πίνακα A[12, 4].
- γ. Θα διαβάζει τις ποσότητες σε κιλά που χρησιμοποιήθηκαν από το κάθε ένα υλικό για κάθε ένα μήνα του έτους 2009 στο εργοστάσιο Β, και να τα τοποθετεί στον πίνακα B[12, 4].
- δ. Θα ελέγχει και θα εμφανίζει το όνομα του εργοστασίου που χρησιμοποίησε περισσότερη ποσότητα νερού για ένα έτος για την παραγωγή χυμών.
- ε. Για τον μήνα Απρίλιο, θα βρίσκει και θα εμφανίζει την ονομασία του φρούτου (ή των φρούτων) που χρησιμοποιήθηκε περισσότερο ως πρώτη ύλη στο εργοστάσιο Β.

Παρατήρηση: Οι ποσότητες από το κάθε υλικό καταχωρίζονται ως ακέραιοι αριθμοί.

Στοιβά (Stack)

Στη στοιβά τα δεδομένα στοιβάζονται το ένα πάνω στο άλλο όπως σε μια στοιβά από πιάτα. Η στοιβά υλοποιεί τη λογική LIFO (*Last In First Out*). Τα δεδομένα εισάγονται στην κορυφή της στοιβάς ενώ η αφαίρεση ενός στοιχείου γίνεται πάντα από την κορυφή της στοιβάς.

Στους αλγορίθμους μπορούμε να χρησιμοποιήσουμε έναν μονοδιάστατο πίνακα για να προσομοιώσουμε τις λειτουργίες της στοιβάς. Η στατική στοιβά υλοποιείται με μονοδιάστατο πίνακα N θέσεων και χρησιμοποιεί ένα **δείκτη Top** που δείχνει στην κορυφή της στοιβάς.

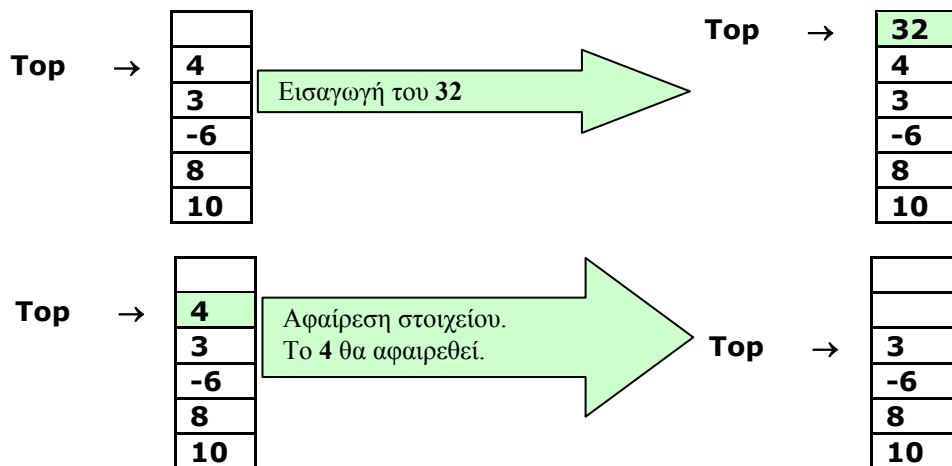
Οι λειτουργίες είναι δύο :

- Η **εισαγωγή-ώθηση (Push)** ενός στοιχείου στην κορυφή της στοιβάς. Στη διαδικασία πρέπει να ελέγχεται αν η στοιβά είναι γεμάτη οπότε λέγεται ότι συμβαίνει υπερχείλιση (overflow) της στοιβάς.
- Η **αφαίρεση-απόθεση (Pop)** ενός στοιχείου από την κορυφή της στοιβάς. Ελέγχεται αν υπάρχει ένα τουλάχιστον στοιχείο στη στοιβά, δηλαδή ελέγχει αν γίνεται υποχείλιση (underflow) της στοιβάς.

Παρατηρήσεις:

1. Η μέθοδος επεξεργασίας της στοιβάς ονομάζεται τελευταίο μέσα πρώτο έξω (LIFO).
2. Κάθε νέο στοιχείο προστίθεται μόνο στην κορυφή της στοιβάς.
3. κάθε φορά μπορούμε να επεξεργαστούμε μόνο το στοιχείο που βρίσκεται στην κορυφή της στοιβάς.
4. Για να επεξεργαστούμε ένα στοιχείο που δεν βρίσκεται στην κορυφή πρέπει να απωθήσουμε τα στοιχεία που βρίσκονται από πάνω του.
5. Μπορεί να γίνει απόθεση από την κορυφή της στοιβάς ενός μόνο στοιχείου τη φορά.

Παράδειγμα:



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

Αλγόριθμος υλοποίησης λειτουργιών ώθησης/απόθησης

Έστω για την υλοποίηση μιας στοίβας χρησιμοποιούμε:

1. τον μονοδιάστατο πίνακα table με N θέσεις
2. τη μεταβλητή top (περιέχει τη θέση του στοιχείου του πίνακα, στο οποίο υπάρχει η κορυφή της στοίβας)

Α. Ώθηση ενός δεδομένου

Αν υπάρχει διαθέσιμος χώρος ($top < N$, έλεγχος για υπερχείλιση), αυξάνεται η μεταβλητή top κατά ένα και στη θέση του πίνακα που δείχνει, τοποθετείται το X. Αν δεν υπάρχει διαθέσιμος χώρος (ΑΛΛΙΩΣ), η στοίβα είναι γεμάτη.

```
ΔΙΑΒΑΣΕ X
ΑΝ top < N ΤΟΤΕ
    top ← top + 1
    table[top] ← X
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'στοίβα γεμάτη'
ΤΕΛΟΣ_ΑΝ
```

Β. Ώθηση πολλών δεδομένων

Το παραπάνω τμήμα εντολών το τοποθετούμε μέσα σε μία επαναληπτική διαδικασία μέχρι η στοίβα να γεμίσει ($top = N$)

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ X
ΑΝ top < N ΤΟΤΕ
    top ← top + 1
    table[top] ← X
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'στοίβα γεμάτη'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ top = N
```

Γ. Απόθηση ενός δεδομένου

Αν υπάρχει έστω και ένα στοιχείο στη στοίβα ($top \geq 1$, έλεγχος για υποχείλιση), εξάγεται αυτό που είναι στη κορυφή και μειώνεται η μεταβλητή top κατά ένα. Αν δεν υπάρχουν στοιχεία (ΑΛΛΙΩΣ), η στοίβα είναι άδεια.

```
ΑΝ top ≥ 1 ΤΟΤΕ
    X ← table[top]
    top ← top - 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Στοιβα άδεια'
ΤΕΛΟΣ_ΑΝ
```

Δ. Απόθηση πολλών δεδομένων

Το παραπάνω τμήμα εντολών το τοποθετούμε μέσα σε μία επαναληπτική διαδικασία μέχρι η στοίβα να αδειάσει ($top = 0$)

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ top ≥ 1 ΤΟΤΕ
    X ← table[top]
    top ← top - 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Στοιβα άδεια'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ top = 0
```

Σχόλια:

- Στην απόθεση δεν διαγράφεται το στοιχείο, στην πραγματικότητα δε γίνεται καμία παρέμβαση στον πίνακα. Ο δείκτης κορυφή δείχνει σε άλλη θέση και το υπό απόθεση στοιχείο, το πρώην τελευταίο στη στοίβα, δεν υπόκειται σε επεξεργασία.
- Ο δείκτης κορυφή καθορίζει το ενεργό μέρος του πίνακα που υποβάλλεται σε επεξεργασία.

Ουρά (Queue)

Στη δομή δεδομένων ουρά τα δεδομένα παρατάσσονται το ένα πίσω από το άλλο σχηματίζοντας μια ουρά, όπως μια ουρά στην τράπεζα.

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 **Front** που δείχνει στο μπροστινό μέρος της ουράς και τον **Rear** που δείχνει στο πίσω μέρος της ουράς.

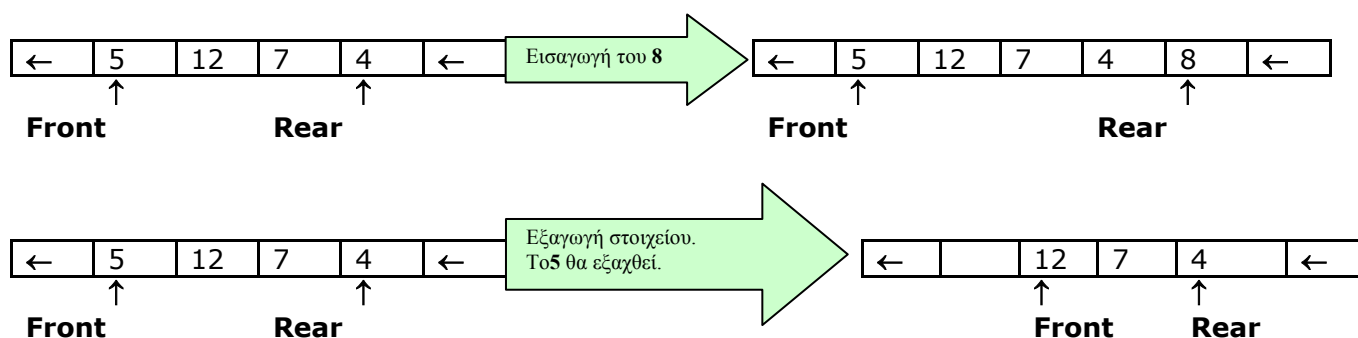
Οι λειτουργίες είναι δύο :

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

Παρατηρήσεις

1. Κάθε στοιχείο προστίθεται στο τέλος της ουράς.
2. Κάθε φορά μπορούμε να επεξεργαστούμε μόνο το στοιχείο που βρίσκεται στην αρχή της ουράς.
3. Μπορεί να γίνει εξαγωγή από την αρχή της ουράς, μόνο ενός στοιχείου κάθε φορά.
4. Για να επεξεργαστούμε ένα στοιχείο που δεν βρίσκεται στην αρχή της ουράς πρέπει να εξάγουμε όλα τα στοιχεία που βρίσκονται μπροστά του.

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.

Αλγόριθμος υλοποίησης λειτουργιών εισαγωγής/εξαγωγής

Έστω για την υλοποίηση μιας ουράς χρησιμοποιούμε:

1. τον μονοδιάστατο πίνακα table με N θέσεις
2. τη μεταβλητή front (περιέχει τη θέση του στοιχείου του πίνακα, στο οποίο υπάρχει το εμπρός άκρο της ουράς)
3. τη μεταβλητή rear (περιέχει τη θέση του στοιχείου του πίνακα, στο οποίο υπάρχει το πίσω άκρο της ουράς)

A. Εισαγωγή ενός δεδομένου

Αν υπάρχει διαθέσιμος χώρος ($rear < N$, έλεγχος για υπερχείλιση), αυξάνεται η μεταβλητή rear κατά ένα και στη θέση του πίνακα που δείχνει, τοποθετείται το X. Αν δεν υπάρχει διαθέσιμος χώρος (ΑΛΛΙΩΣ), η ουρά είναι γεμάτη στο «πίσω μέρος».


```

ΔΙΑΒΑΣΕ Χ
ΑΝ rear<N ΤΟΤΕ
    rear←rear + 1
    table[rear] ←Χ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ουρά γεμάτη στο πίσω μέρος'
ΤΕΛΟΣ_ΑΝ

```

Β. Εισαγωγή πολλών δεδομένων

Το παραπάνω τμήμα εντολών το τοποθετούμε μέσα σε μία επαναληπτική διαδικασία μέχρι η ουρά να γεμίσει (rear=N)

```

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Χ
ΑΝ rear<N ΤΟΤΕ
    rear←rear + 1
    table[rear] ←Χ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ουρά γεμάτη στο πίσω μέρος'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ rear=N

```

Γ. Εξαγωγή ενός δεδομένου

Αν η ουρά περιέχει στοιχεία (front <= rear, έλεγχος για υποχείλιση), εξάγεται αυτό που είναι στην αρχή και αυξάνεται η μεταβλητή front κατά ένα. Αν δεν υπάρχουν στοιχεία (ΑΛΛΙΩΣ), η ουρά είναι άδεια.

```

ΑΝ front <= rear ΤΟΤΕ
    Χ←table[front]
    front←front + 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Ουρά άδεια'
ΤΕΛΟΣ_ΑΝ

```

Δ. Εξαγωγή πολλών δεδομένων

Το παραπάνω τμήμα εντολών το τοποθετούμε μέσα σε μία επαναληπτική διαδικασία μέχρι η ουρά να αδειάσει (front > rear)

```

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ front <= rear ΤΟΤΕ
    Χ←table[front]
    front←front + 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Ουρά άδεια'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ front > rear

```

Λυμένα Παραδείγματα

41. Δίνεται η διπλανή στοίβα. Να σχεδιάσετε την κατάσταση της στοίβας και να δείξετε την τιμή της μεταβλητής top μετά από την εκτέλεση της κάθε μιας από τις λειτουργίες: α. ώθηση του στοιχείου 2. β. ώθηση του στοιχείου 9. γ. απώθηση. δ. απώθηση

5	
4	
3	-4
2	8
1	5

5		5	9 top=5	5		5	
4	2 top=4	4	2	4	2 top=4	4	
3	-4	3	-4	3	-4	3	-4 top=3
2	8	2	8	2	8	2	8
1	5	1	5	1	5	1	5

42. Δίνεται εξής ακολουθία γραμμάτων: C, D, E, A, F, H. α. Να τοποθετήσετε τα γράμματα σε μια ουρά 10 θέσεων και να σχεδιάσετε την ουρά. Ποιες τιμές θα έχουν οι δείκτες της ουράς. β. Να τοποθετήσετε τα γράμματα σε μια στοίβα 8 θέσεων και να σχεδιάσετε τη στοίβα. Ποια θα είναι η τιμή του δείκτη της στοίβας. γ. Να σχεδιάσετε την ουρά μετά τη διαδοχική εξαγωγή δύο στοιχείων. Ποια στοιχεία θα εξαχθούν; Να σχεδιάσετε τη στοίβα, μετά την εισαγωγή του στοιχείου B. δ. Πόσες φορές πρέπει να εκτελεστεί η λειτουργία της απώθησης, για να εξαχθεί το γράμμα A από τη στοίβα, και πόσες φορές πρέπει να εκτελεστεί η λειτουργία της εξαγωγής, για να εξαχθεί το γράμμα F από την ουρά.

α.

C	D	E	A	F	H				
---	---	---	---	---	---	--	--	--	--

εμπρός =1

πίσω=6

β.

H	top=6
F	
A	
E	
D	
C	

γ1.

		E	A	F	H				
--	--	---	---	---	---	--	--	--	--

εμπρός =3

πίσω=6

γ2.

B	top=7
H	
F	
A	
E	
D	
C	

δ. Για να εξέλθει το στοιχείο A από τη στοίβα, πρέπει να εκτελεστεί 4 φορές η λειτουργία της απώθησης. Για να εξέλθει το στοιχείο F από την ουρά, πρέπει να εκτελεστεί η λειτουργία της εξαγωγής 3 φορές.

43. 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 και να σχεδιάσετε την παραπάνω στοίβα.

ii. Αν εφαρμόσετε τις παρακάτω λειτουργίες: Απώθηση, Απώθηση, Απώθηση, Ώθηση Χ, Ώθηση Δ και Απώθηση, ποια είναι η νέα τιμή της top και ποια η τελική μορφή της στοίβας;

2) Σε μια άδεια στοίβα 10 θέσεων ωθούνται τα στοιχεία Ο, Σ, Λ, Τ, Ε. Με ποιον τρόπο πρέπει να γίνει η ώθηση και η απώθηση των στοιχείων, ώστε να έχουμε ως έξοδο τα στοιχεία Τ, Ε, Λ, Ο, Σ, με το στοιχείο Σ να βρίσκεται στην κορυφή της στοίβας;

1)

Δ TOP=5	
Α	
Μ	Χ TOP=3
Γ	Γ
Σ	Σ

2)

Ε TOP=5	Σ TOP=5
Τ	Ο
Λ	Λ
Σ	Ε
Ο	Τ

Εκτελώντας τις λειτουργίες : Απώθηση, Απώθηση, Απώθηση, Απώθηση, Απώθηση, Ώθηση Τ, Ώθηση Ε, Ώθηση Λ, Ώθηση Ο, Ώθηση Σ

44. 1. Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.

i. Να προσδιορίσετε τις τιμές των δεικτών rear και front και να σχεδιάσετε την παραπάνω ουρά.

ii. Αν εφαρμόσετε τις παρακάτω λειτουργίες: Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή Χ, Εισαγωγή Δ και Εξαγωγή, ποιες είναι οι τιμές των δεικτών rear και front της ουράς και η τελική μορφή της;

2. Σε μια άδεια ουρά 10 θέσεων εισάγονται τα στοιχεία Ο, Σ, Λ, Τ, Ε. Με ποιον τρόπο πρέπει να εισαχθούν και να εξαχθούν τα στοιχεία, ώστε η έξοδος να εμφανίζει τα στοιχεία Τ, Ε, Λ, Ο, Σ;

1)

Σ	Γ	Μ	Α	Δ						
---	---	---	---	---	--	--	--	--	--	--

front=1 και rear=5

Η τελική μορφή της ουράς είναι:

				Δ	Χ	Δ				
--	--	--	--	---	---	---	--	--	--	--

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

2)

Η αρχική μορφή της ουράς είναι:

Ο	Σ	Λ	Τ	Ε					
---	---	---	---	---	--	--	--	--	--

front=1 και rear=5

Με την εκτέλεση των λειτουργιών: Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή Λ, Εισαγωγή Ο, Εισαγωγή Σ, η τελική μορφή της ουράς γίνεται:

			Τ	Ε	Λ	Ο	Σ		
--	--	--	---	---	---	---	---	--	--

και οι τιμές της front και της rear γίνονται: front=4 και rear=8

45. Ένα πρότυπο πλυντήριο πιάτων έχει την ιδιότητα να δέχεται μια στοίβα το πολύ 100 πιάτων και η λειτουργία του καθορίζεται από δύο αριθμημένα κουμπιά, με τους αριθμούς 1 και 2. Ο χειριστής εισάγει την αρχική στοίβα πιάτων στο πλυντήριο και πιέζει το κουμπί με τον αριθμό 2, ώστε πλυντήριο να αφαιρέσει το πάνω πιάτο, για να το πλένει. Όταν ο χειριστής θέλει να εισάγει ένα πιάτο, τότε πιέζει το κουμπί με τον αριθμό 1, οπότε και εισάγεται ένα πιάτο στο πάνω μέρος της στοίβας. Να γράφει πρόγραμμα που θα υλοποιεί το εξής: α. Χρησιμοποιεί τον πίνακα A[100] για να προσομειώνει τη στοίβα των πιάτων. β. Διαβάζει τον αριθμό των πιάτων της αρχικής στοίβας που τοποθετείται στο πλυντήριο εξασφαλίζοντας ότι είναι ένας αριθμός από το 1-100. Για κάθε ένα από αυτά εισάγει το γράμμα «Π» στον πίνακα A. γ. Διαβάζει επαναληπτικά τον αριθμό του κουμπιού που πιέζει ο χρήστης. Αν είναι το 2, τότε με την ενέργεια της απόθησης, εφόσον υπάρχουν διαθέσιμα πιάτα, αφαιρείται προς πλύσιμο το πάνω και εμφανίζεται το μήνυμα «ΠΛΥΣΗ», διαφορετικά το πρόγραμμα τερματίζει καθώς θα έχουν πλυθεί όλα τα πιάτα. Αν είναι το 1, τότε αν υπάρχει χώρος, με την ενέργεια της ώθησης εισάγεται ένα νέο πιάτο (το γράμμα «Π» στην υπάρχουσα στοίβα και εμφανίζεται το μήνυμα «ΝΕΟ ΠΙΑΤΟ», διαφορετικά το μήνυμα «ΔΕΝ ΥΠΑΡΧΕΙ ΧΩΡΟΣ». δ. Εμφανίζει: 1. τον συνολικό αριθμό των πιάτων που πλύθηκαν και 2. το μήνυμα «Δεν τοποθετήθηκαν νέα πιάτα» αν δεν εισήχθησαν νέα πιάτα στο πλυντήριο, διαφορετικά εμφανίζει πόσες φορές υπήρχε διαδοχική εισαγωγή τριών νέων πιάτων.

ΠΡΟΓΡΑΜΜΑ ΠΛΥΝΤ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ΤΟΡ, Ι, ΠΙΑΤΑ, ΠΛΗΘ, ΚΟΥΜΠΙ, ΠΛΗΘ_ΔΙΑΔ, ΔΙΑΔ, ΠΛΗΘ_ΝΕΩΝ
ΧΑΡΑΚΤΗΡΕΣ: A[100], Y

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ΠΙΑΤΑ

ΜΕΧΡΙΣ_ΟΤΟΥ ΠΙΑΤΑ >= 1 ΚΑΙ ΠΙΑΤΑ <= 100

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ ΠΙΑΤΑ

A[I] ← 'Π'

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΠΛΗΘ ← 0

ΠΛΗΘ_ΝΕΩΝ ← 0

ΔΙΑΔ ← 0

ΠΛΗΘ_ΔΙΑΔ ← 0

ΤΟΡ ← ΠΙΑΤΑ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ΚΟΥΜΠΙ

ΑΝ ΚΟΥΜΠΙ = 1 ΤΟΤΕ

ΑΝ ΤΟΡ < 100 ΤΟΤΕ

ΤΟΡ ← ΤΟΡ + 1

A[ΤΟΡ] ← 'Π'

ΓΡΑΨΕ 'ΝΕΟ ΠΙΑΤΟ'

ΠΛΗΘ_ΝΕΩΝ ← ΠΛΗΘ_ΝΕΩΝ + 1

ΔΙΑΔ ← ΔΙΑΔ + 1

```

        AN ΔΙΑΔ=3 ΤΟΤΕ
            ΠΛΗΘ_ΔΙΑΔ←ΠΛΗΘ_ΔΙΑΔ+1
            ΔΙΑΔ←0
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'ΔΕΝ ΥΠΑΡΧΕΙ ΧΩΡΟΣ'
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
    AN ΤΟΡ>=1 ΤΟΤΕ
        Υ←Α[ΤΟΡ]
        ΤΟΡ←ΤΟΡ-1
        ΠΛΗΘ←ΠΛΗΘ+1
        ΔΙΑΔ←0
        ΓΡΑΨΕ 'ΠΛΥΣΗ'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ ΤΟΡ=0
ΓΡΑΨΕ ΠΛΗΘ
ΑΝ ΠΛΗΘ_ΝΕΩΝ=0 ΤΟΤΕ
    ΓΡΑΨΕ 'ΔΕΝ ΤΟΠΟΘΕΤΗΘΗΚΑΝ ΝΕΑ ΠΙΑΤΑ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ ΠΛΗΘ_ΔΙΑΔ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΠΛΥΝΤ

```

46. Μια υπάλληλος του ΟΑΕΔ εξυπηρετεί ανέργους με σειρά προτεραιότητας. Ο κάθε άνεργος προμηθεύεται έναν αριθμό που εκδίδεται από ειδικό μηχάνημα σχηματίζοντας έτσι μια σειρά προτεραιότητας. Ο αριθμός προτεραιότητας αρχίζει από το νούμερο 100 και αυξάνεται για κάθε άνεργο που επιθυμεί να εξυπηρετηθεί από την υπάλληλο. Η έκδοση των αριθμών διακόπτεται από το μηχάνημα όταν φτάσει στον αριθμό 150. Να γραφεί πρόγραμμα που:

- α. Χρησιμοποιεί τον πίνακα KIN[50] για να προσομειώνει την ουρά εξυπηρέτησης των ανέργων.**
- β. Δημιουργεί και καταχωρίζει στον πίνακα KIN τους αριθμούς προτεραιότητας των 15 πρώτων ανέργων.**
- γ. Διαβάζει επαναληπτικά το γράμμα «N», που καθορίζει την εισοδο νέου ανέργου ή το γράμμα «E» που δηλώνει ότι ένας άνεργος θα εξυπηρετηθεί και πραγματοποιεί τις παρακάτω ενέργειες:**
 - 1. Κατά την εισοδο ενός νέου ανέργου εξετάζει αν το μηχάνημα μπορεί να εκδώσει αριθμό προτεραιότητας, τον οποίο δημιουργεί και καταχωρίζει στην ουρά με την λειτουργία της εισαγωγής, διαφορετικά θα εμφανίζει το μήνυμα «Δεν μπορείτε να λάβετε αριθμό προτεραιότητας».**
 - 2. όταν ένας άνεργος πρόκειται να εξυπηρετηθεί, εξετάζεται αν υπάρχουν τέτοιοι άνεργοι και στην συνέχεια με την λειτουργία της εξαγωγής αφαιρείται από την ουρά ο αριθμός προτεραιότητάς του και εμφανίζεται το μήνυμα «Εξυπηρέτηση».** Η επαναληπτική διαδικασία ολοκληρώνεται όταν εξυπηρετηθούν όλοι οι άνεργοι.
 - δ. Εμφανίζει:**
 - 1. το συνολικό αριθμό των ανέργων που εξυπηρετήθηκαν.**
 - 2. πόσες φορές η ουρά είχε μόνο έναν άνεργο και**
 - 3. τον αριθμό προτεραιότητας του τελευταίου ανέργου που εξυπηρετήθηκε.**

```

ΠΡΟΓΡΑΜΜΑ ΟΑΕΔ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, FRONT, REAR, ΠΛΗΘ_ΑΝ, ΠΛΗΘ_ΕΝΑ, ΑΡΙΘ, KIN[50], Y, ΟΥΡΑ[15]
    ΧΑΡΑΚΤΗΡΕΣ: ΕΝΕΡ
ΑΡΧΗ
    ΠΛΗΘ_ΑΝ←0
    ΠΛΗΘ_ΕΝΑ←0
    ΑΡΙΘ←99
    ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 15
        ΑΡΙΘ←ΑΡΙΘ+1
        KIN[I]←ΑΡΙΘ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    FRONT←1

```

```

REAR←15
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ ΕΝΕΡ
  ΑΝ ΕΝΕΡ='Ν' ΤΟΤΕ
    ΑΝ REAR<50 ΤΟΤΕ
      ΑΡΙΘ←ΑΡΙΘ+1
      REAR←REAR+1
      ΟΥΡΑ[REAR]←ΑΡΙΘ
    ΑΛΛΙΩΣ
      ΓΡΑΨΕ 'ΔΕΝ ΛΑΜΒΑΝΕΤΕ ΑΡΙΘΜ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ'
  ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
  ΑΝ FRONT<=REAR ΤΟΤΕ
    Υ←ΟΥΡΑ[FRONT]
    ΓΡΑΨΕ 'ΕΞΥΠΗΡΕΤΗΣΗ'
    FRONT←FRONT+1
    ΠΛΗΘ_ΑΝ←ΠΛΗΘ_ΑΝ+1
  ΤΕΛΟΣ_ΑΝ
  ΑΝ FRONT=REAR ΤΟΤΕ
    ΠΛΗΘ_ΕΝΑ←ΠΛΗΘ_ΕΝΑ+1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ FRONT>REAR
ΓΡΑΨΕ ΠΛΗΘ_ΑΝ, ΠΛΗΘ_ΕΝΑ, Υ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΟΑΕΔ

```

47. Ένα οχηματαγωγό πλοίο, χωρητικότητας 250 αυτοκινήτων, τα οποία δύνανται να τοποθετηθούν αποκλειστικά σε μία σειρά, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Στο λιμάνι του Πειραιά προσέρχονται τα οχήματα για αναχώρηση. Τα οχήματα που επιβιβάζονται πρώτα είναι αυτά που θα αποβιβάστούν τελευταία.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υλοποιεί μενού με τις επιλογές:

1. Επιβίβαση 2. Αποβίβαση 3. Έξοδος

2. Στην περίπτωση που επιλεγθεί η Επιβίβαση, να ζητείται εισαγωγή του αριθμού κυκλοφορίας καθενός από τα οχήματα που προσέρχονται και ο αριθμός κυκλοφορίας του να καταχωρείται στη στοιβα ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα όχημα να τυπώνεται το ερώτημα «Υπάρχει όχημα για επιβίβαση; (N/O)». Αν ο χρήστης απαντήσει Ν (=ΝΑΙ), τότε να επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει Ο (=ΟΧΙ), τότε να σταματά η διαδικασία επιβίβασης και το πρόγραμμα να επιστρέφει στο μενού Επιλογής.

3. Στην περίπτωση που επιλεγθεί η Αποβίβαση, να τυπώνει τον αριθμό κυκλοφορίας όλων των οχημάτων με τη σειρά που αποβιβάζονται από το πλοίο στην ΑΙΓΙΝΑ.

4. Στο τέλος να τυπώνει το πλήθος των οχημάτων που επιβιβάστηκαν στο λιμάνι του ΠΕΙΡΑΙΑ.

ΠΡΟΓΡΑΜΜΑ πλοίο

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ1, πλ2

ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]

ΑΡΧΗ

τοπ <- 0

πλ1 <- 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

πλ2 <- 0

ΓΡΑΨΕ 'Μενού Επιλογών'

ΓΡΑΨΕ '1. Επιβίβαση'

ΓΡΑΨΕ '2. Αποβίβαση'

ΓΡΑΨΕ '3. Έξοδος'

```

ΓΡΑΨΕ 'Δώσε επιλογή:'
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ επ1
  ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
    ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
  ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
ΑΝ επ1 = 1 ΤΟΤΕ
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (N/O);'
      ΔΙΑΒΑΣΕ επ2
      ΑΝ επ2<>'N' ΚΑΙ επ2<>'n' ΚΑΙ επ2<>'O' ΚΑΙ επ2<>'o' ΤΟΤΕ
        ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
      ΤΕΛΟΣ_ΑΝ
      ΜΕΧΡΙΣ_ΟΤΟΥ επ2='O' Η επ2='o' Η επ2='N' Η επ2='n'
      ΑΝ επ2 = 'N' Η επ2 = 'n' ΤΟΤΕ
        ΑΝ τοπ < 250 ΤΟΤΕ
          ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
          ΔΙΑΒΑΣΕ αρ
          τοπ <- τοπ + 1
          π[τοπ] <- αρ
          πλ1 <- πλ1 + 1
          ΑΝ τοπ = 250 ΤΟΤΕ
            ΓΡΑΨΕ 'Το πλοίο δεν χωρά άλλα οχήματα'
          ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
      ΤΕΛΟΣ_ΑΝ
    ΜΕΧΡΙΣ_ΟΤΟΥ τοπ = 250 Η επ2 = 'O' Η επ2 = 'o'
  ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
    ΟΣΟ τοπ >= 1 ΕΠΑΝΑΛΑΒΕ
      ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[τοπ]
      τοπ <- τοπ - 1
      πλ2 <- πλ2 + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ ', πλ2
  ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
ΓΡΑΨΕ 'Οχήματα που επιβιβάστηκαν στον ΠΕΙΡΑΙΑ:', πλ1
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο

```

48. Ένα οχηματαγωγό πλοίο με δύο διαφορετικές πόρτες, μία για την είσοδο και μία για την έξοδο των οχημάτων, χωρητικότητας 250 αυτοκινήτων, τα οποία δύνανται να τοποθετηθούν αποκλειστικά σε μία σειρά, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Τα οχήματα που επιβιβάζονται πρώτα είναι και αυτά που θα αποβιβαστούν πρώτα. Στο λιμάνι του Πειραιά προσέρχονται τα αυτοκίνητα για αναχώρηση.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υλοποιεί μενού με τις επιλογές:

1. Επιβίβαση 2. Αποβίβαση 3. Έξοδος

2. Στην περίπτωση που επιλεγεί η Επιβίβαση το πρόγραμμα θα διαβάζει τον αριθμό κυκλοφορίας καθενός από τα οχήματα που επιβιβάζονται στο πλοίο και θα τον καταχωρίζει στην ουρά ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα όχημα να τυπώνεται το ερώτημα «Υπάρχει όχημα για επιβίβαση; (N/O)». Αν ο χρήστης απαντήσει N (=ΝΑΙ), τότε να επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει O (=ΟΧΙ), τότε να σταματά η διαδικασία επιβίβασης και να επιστρέφει το πρόγραμμα στο μενού Επιλογής.

3. Στην περίπτωση που επιλεγεί η Αποβίβαση το πρόγραμμα θα εξαγει από την ουρά και θα

εμφανίζει όλα τα αυτοκίνητα που αποβιβάστηκαν στην ΑΙΓΙΝΑ.

ΠΡΟΓΡΑΜΜΑ πλοίο2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: αρχ, τελος, επ1, πλ

ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]

ΑΡΧΗ

αρχ <- 0

τελος <- 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Μενού Επιλογών'

ΓΡΑΨΕ '1. Επιβίβαση'

ΓΡΑΨΕ '2. Αποβίβαση'

ΓΡΑΨΕ '3. Έξοδος'

ΓΡΑΨΕ 'Δώσε επιλογή:'

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ επ1

ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ

ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3

ΑΝ επ1 = 1 ΤΟΤΕ

ΑΝ τελος = 250 ΤΟΤΕ

ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'

ΑΛΛΙΩΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (N/O);'

ΔΙΑΒΑΣΕ επ2

ΑΝ επ2 <> 'N' ΚΑΙ επ2 <> 'n' ΚΑΙ επ2 <> 'O' ΚΑΙ επ2 <> 'o' ΤΟΤΕ

ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ επ2='O' Η επ2='o' Η επ2='N' Η επ2='n'

ΑΝ επ2 = 'N' Η επ2 = 'n' ΤΟΤΕ

ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'

ΔΙΑΒΑΣΕ αρ

ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ

αρχ <- 1

τελος <- 1

π[τελος] <- αρ

ΑΛΛΙΩΣ

τελος <- τελος + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ τελος = 250 Η επ2 = 'O' Η επ2 = 'o'

ΑΝ τελος = 250 ΤΟΤΕ

ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ

πλ <- 0

ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ

ΓΡΑΨΕ 'Το πλοίο είναι άδειο'

ΑΛΛΙΩΣ_ΑΝ αρχ = τελος ΤΟΤΕ

ΓΡΑΨΕ 'Αποβιβάζεται το μοναδικό όχημα:', π[αρχ]

π[αρχ] <- ''

αρχ <- 0

τελος <- 0

πλ <- πλ + 1

ΑΛΛΙΩΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ


```

        ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[αρχ]
        π[αρχ] <- "
        αρχ <- αρχ + 1
        πλ <- πλ + 1
        ΜΕΧΡΙΣ_ΟΤΟΥ (αρχ > τελος)
    ΤΕΛΟΣ_ΑΝ
    ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ:', πλ
    αρχ <- 0
    τελος <- 0
    ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
ΓΡΑΨΕ 'ΕΞΟΔΟΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο2

```

Ασκήσεις

36. Δίνεται η στοίβα, να σχεδιάσετε την κατάσταση της στοίβας και να δείξετε την τιμή της top μετά από την εκτέλεση: α. Ώθηση του Α, β. Ώθηση του Γ, γ. Απώθηση, δ. Απώθηση, ε. Απώθηση

B top=2
Δ

37. Δίνεται η παρακάτω ακολουθία αριθμών: 25, 8, 12, 14, 71, 41, 1. Τοποθετούμε τους αριθμούς σε στοίβα και σε ουρά.

1. Ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών στη στοίβα και ποια για την τοποθέτησή τους στην ουρά;
2. Να σχεδιάσετε τις δύο δομές (στοίβα και ουρά) μετά την τοποθέτηση των αριθμών. Μονάδες 4
3. Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο αριθμών από τη στοίβα και ποια για την έξοδό τους από την ουρά;
4. Πόσες φορές θα πρέπει να γίνει η παραπάνω λειτουργία στη στοίβα και πόσες στην ουρά για να εξέλθει ο αριθμός 71;

38. Η ουρά είναι μία δομή δεδομένων.

1. Να δώσετε ένα παράδειγμα ουράς από την καθημερινή ζωή.
2. Να αναφέρετε τις λειτουργίες της ουράς και τους δείκτες που απαιτούνται.
3. Σε μία ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Μ, Κ, Δ, Α, Σ στην πρώτη, δεύτερη, τρίτη, τέταρτη και πέμπτη θέση αντίστοιχα.
 - α. Να προσδιορίσετε τις τιμές των δεικτών της παραπάνω ουράς.
 - β. Στη συνέχεια να αφαιρέσετε ένα στοιχείο από την ουρά. Ποιος δείκτης μεταβάλλεται και ποια η νέα του τιμή;
 - γ. Τέλος να τοποθετήσετε το στοιχείο Λ στην ουρά. Ποιος δείκτης μεταβάλλεται και ποια η νέα του τιμή;

39. Μια στοίβα έχει ήδη καταχωρισμένους 5 ακέραιους αριθμούς και αναπαρίσταται από τον πίνακα Α[10]. Να γράψετε τους αριθμούς 1-6 ώστε το τμήμα να διαβάζει έναν αριθμό και να τον τοποθετεί στη στοίβα

```

TOP←...1...
ΔΙΑΒΑΣΕ X
ΑΝ TOP < ....2.... ΤΟΤΕ
    TOP←TOP.....3.....4.....
    Α[.....5.....]←.....6.....

```

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΣΤΟΙΒΑ ΓΕΜΑΤΗ'

ΤΕΛΟΣ_ΑΝ

40. Μια ουρά ακεραίων αντιπροσωπεύεται από πίνακα $O[150]$. Να γραφεί πρόγραμμα που: θα διαβάζει το πλήθος των ακεραίων που υπάρχουν τοποθετημένοι στην αρχή της ουράς, διαβάζει το πολύ 10 ακέραιους αριθμούς και τους τοποθετεί στην ουρά εκτελώντας τη λειτουργία της εισαγωγής. Τερματίζει το διάβασμα όταν έχουν διβαστεί και οι 10 αριθμοί ή όταν η ουρά δεν μπορεί να δεχθεί άλλους, οπότε εμφανίζει το μήνυμα 'Τερματισμός εισαγωγής'.

Ερωτήσεις Σ/Λ

1.

1. Σε μια δομή δεδομένων η διαγραφή αποτελεί την αντίστροφη πράξη της συγχώνευσης. (Επαναληπτικές Ημ 2016) Σ Λ
2. Η εισαγωγή είναι η λειτουργία με την οποία αντικαθιστούμε το περιεχόμενο ενός κόμβου μιας δομής δεδομένων. Σ Λ
3. Σε μία στατική δομή, το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης καθορίζεται κατά την εκτέλεση του προγράμματος. (Ημερήσια 2009) Σ Λ
4. Οι δυναμικές δομές δεδομένων αποθηκεύονται πάντα σε συνεχόμενες θέσεις μνήμης. (Ημερήσια 2012 & Εσπερινά 2012) Σ Λ
5. Σε έναν πίνακα μπορούμε να καταχωρίσουμε τα ονόματα και τις ηλικίες 50 ποδοσφαιριστών. Σ Λ
6. Κάθε στοιχείο ενός πίνακα πρέπει υποχρεωτικά να περιλαμβάνει το όνομα του πίνακα ακολουθούμενο από έναν ή περισσότερους δείκτες μέσα σε αγκύλες. Σ Λ
7. Ο πίνακας είναι μία δομή που μπορεί να περιέχει στοιχεία διαφορετικού τύπου. (Ημερήσια 2011 & Εσπερινά 2011) Σ Λ
8. Οι στατικές δομές στηρίζονται στην τεχνική της δυναμικής παραχώρησης μνήμης. (Ημερίσια 2016 -Νέο) Σ Λ
9. Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος. (Ημερήσια 2018) Σ Λ

2.

1. Με την εντολή $A[100] \leftarrow 100$ εκχωρούμε τον αριθμό 100 στο 100ο στοιχείο του πίνακα A. Σ Λ
2. Ο δείκτης ενός πίνακα μπορεί να είναι πραγματικός αριθμός. Σ Λ
3. Μπορούμε στο εσωτερικό ενός προγράμματος να μεταβάλλουμε το περιεχόμενο του κάθε στοιχείου ενός πίνακα $A[10]$ του, αλλά όχι και το μέγεθος του. Σ Λ
4. Ο πίνακας είναι μια στατική δομή δεδομένων. Σ Λ
5. Τα στοιχεία ενός μονοδιάστατου πίνακα αποθηκεύονται πάντα σε συνεχόμενες θέσεις μνήμης. Σ Λ
6. Σε ένα μονοδιάστατο πίνακα, η αναφορά σε κάθε στοιχείο του, γίνεται με το όνομα του πίνακα ακολουθούμενο από δύο δείκτες. Σ Λ
7. Η δομή δεδομένων «Πίνακας» στη γλώσσα μπορεί να περιέχει και δεδομένα διαφορετικού τύπου. Σ Λ
8. Η δήλωση του τύπου του πίνακα σε ένα πρόγραμμα γραμμένο σε ΓΛΩΣΣΑ, γίνεται υποχρεωτικά στο τμήμα δήλωσης μεταβλητών του προγράμματος. Σ Λ
9. Η ΓΛΩΣΣΑ υποστηρίζει μόνο δύο διαστάσεων πίνακες. Σ Λ
10. Τα στοιχεία ενός πίνακα στη ΓΛΩΣΣΑ αποθηκεύονται σε συνεχόμενες θέσεις μνήμης. Σ Λ
11. Σε έναν πίνακα αριθμών $A[10]$ της ΓΛΩΣΣΑΣ το πρώτο στοιχείο είναι το $A[0]$. Σ Λ
12. Η χρήση των πινάκων σε ένα πρόγραμμα αυξάνει την απαιτούμενη μνήμη. (Ημερήσια 2012) Σ Λ

3.

- | | | |
|--|---|---|
| 1. Όλες οι λειτουργίες που εφαρμόζονται σε μία δομή δεδομένων μπορούν να εφαρμοστούν στη δομή δεδομένων που ονομάζεται «Πίνακας». | Σ | Λ |
| 2. Η «συγχώνευση» ως μία λειτουργία που μπορεί να εφαρμοστεί σε μια δομή δεδομένων, μπορεί να εφαρμοστεί και σε ένα μονοδιάστατο πίνακα. | Σ | Λ |
| 3. Η «εισαγωγή» ως μία λειτουργία που μπορεί να εφαρμοστεί σε μία δομή δεδομένων, δεν μπορεί να εφαρμοστεί σε ένα μονοδιάστατο πίνακα. | Σ | Λ |
| 4. Η εύρεση του μικρότερου στοιχείου ενός μονοδιάστατου πίνακα σημαίνει την εύρεση του αριθμού του στοιχείου του πίνακα, όπου υπάρχει ο μικρότερος αριθμός του πίνακα. | Σ | Λ |
| 5. Μία από τις τυπικές επεξεργασίες ενός πίνακα είναι και η εύρεση του μέγιστου ή του ελάχιστου στοιχείου. | Σ | Λ |
| 6. Σε έναν πίνακα μπορεί να βρεθεί μόνο ένα μικρότερο στοιχείο του. | Σ | Λ |

4.

- | | | |
|--|---|---|
| 1. Όταν ταξινομούμε ένα σύνολο στοιχείων κατά φθίνουσα σειρά, τότε αυτά τοποθετούνται από το μικρότερο προς το μεγαλύτερο. | Σ | Λ |
| 2. Η λειτουργία της ταξινόμησης μπορεί να εφαρμοστεί και σε έναν μονοδιάστατο πίνακα αλφαριθμητικών. | Σ | Λ |
| 3. Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο γρήγορος αλγόριθμος ταξινόμησης. (Επαναληπτικές Ημ. 2006) | Σ | Λ |
| 4. Η μόνη ενέργεια της εντολής «Αντιμετάθεσε X ,Y» είναι να δίνει στο X την τιμή που έχει το Y. | Σ | Λ |
| 5. Με τη μέθοδο της φυσαλίδας μπορούμε να διατάξουμε τα στοιχεία ενός μονοδιάστατου πίνακα μόνο από το μικρότερο προς το μεγαλύτερο. | Σ | Λ |
| 6. Η ταξινόμηση των στοιχείων ενός πίνακα με τη μέθοδο της φυσαλίδας βασίζεται στην αρχή της σύγκρισης και αντιμετάθεσης ζευγών γειτονικών στοιχείων του πίνακα. (Εσπερινά 2010) | Σ | Λ |
| 7. Η λειτουργία της ταξινόμησης σε έναν πίνακα έχει έννοια όταν τα στοιχεία του δεν είναι όλα ίδια μεταξύ τους. | Σ | Λ |
| 8. Όταν στον αλγόριθμο της ταξινόμησης συγκρίνεται το προηγούμενο στοιχείο του πίνακα να είναι μεγαλύτερο από το επόμενο, τότε μιλάμε για αύξουσα ταξινόμηση του πίνακα. | Σ | Λ |
| 9. Στον αλγόριθμο της φυσαλίδας, όταν μετά την εκτέλεση του εσωτερικού βρόχου δεν έχει αλλάξει θέση κανένα στοιχείο, τότε ο πίνακας έχει ταξινομηθεί. | Σ | Λ |
| 10. Η λειτουργία της ταξινόμησης μπορεί να εφαρμοστεί σε έναν πίνακα όταν ορισμένα στοιχεία του είναι ίδια. | Σ | Λ |

5.

- | | | |
|---|---|---|
| 1. Η μέθοδος «ταξινόμηση με επιλογή» μπορεί να εφαρμοστεί και σε έναν μονοδιάστατο πίνακα αλφαριθμητικών. | Σ | Λ |
| 2. Η «ταξινόμηση με επιλογή», βασίζεται στην επιλογή του μικρότερου ή του μεγαλύτερου στοιχείου από αυτά που δεν έχουν ταξινομηθεί και την τοποθέτησή του στην αρχή του πίνακα. | Σ | Λ |
| 3. Για την αύξουσα ταξινόμηση μονοδιάστατου πίνακα με τη μέθοδο «ταξινόμηση με επιλογή» επιλέγεται κάθε φορά το μεγαλύτερο στοιχείο και τοποθετείται στην αρχή του πίνακα. | Σ | Λ |
| 4. Οι δύο αλγόριθμοι ταξινόμησης, της «ευθείας ανταλλαγής» και «με επιλογή» πραγματοποιούν τον ίδιο αριθμό επαναλήψεων για να ταξινομήσουν έναν πίνακα αριθμητικών στοιχείων. | Σ | Λ |
| 5. Με τη μέθοδο της «ταξινόμηση με επιλογή» μπορούμε να διατάξουμε τα στοιχεία ενός μονοδιάστατου πίνακα μόνο από το μικρότερο προς το μεγαλύτερο. | Σ | Λ |
| 6. Στον αλγόριθμο «ταξινόμηση με επιλογή» εκτός από την αντιμετάθεση του περιεχομένου των στοιχείων πίνακα, πραγματοποιείται και αντιμετάθεση των θέσεων τους. | Σ | Λ |
| 7. Για φθίνουσα ταξινόμηση ενός πίνακα με τη μέθοδο «ταξινόμηση με επιλογή», επιλέγεται κάθε φορά το μεγαλύτερο στοιχείο του πίνακα για να μετακινηθεί στις αρχικές θέσεις του. | Σ | Λ |
| 8. Στην «ταξινόμηση με επιλογή» όσο είναι το πλήθος των στοιχείων του πίνακα που ταξινομείται, τόσες αντιμεταθέσεις στοιχείων θα πραγματοποιηθούν. | Σ | Λ |

6.

- | | | |
|--|---|---|
| 1. Όταν η κύρια μνήμη δεν επαρκεί για την αποθήκευση των δεδομένων μιας εφαρμογής, τότε αυτά αποθηκεύονται στη δευτερεύουσα μνήμη. | Σ | Λ |
| 2. Τα δεδομένα ενός πίνακα διατηρούνται, ενώ τα δεδομένα ενός αρχείου χάνονται μετά τον τερματισμό του Η/Υ. | Σ | Λ |
| 3. Οι δομές δεδομένων που αποθηκεύονται στην κύρια μνήμη ενός υπολογιστή είναι γνωστές ως αρχεία. | Σ | Λ |
| 4. Κάθε αρχείο αποτελείται από ένα πλήθος στοιχείων που ονομάζονται εγγραφές. | Σ | Λ |
| 5. Κάθε εγγραφή ενός αρχείου αποτελείται μόνο από πεδία που είναι πρωτεύονται κλειδιά. | Σ | Λ |
| 6. Κάθε εγγραφή ενός αρχείου έχει υποχρεωτικά ένα ή περισσότερα πεδία που την ταυτοποιούν. | Σ | Λ |
| 7. Ένα πεδίο που λειτουργεί ως δευτερεύον κλειδί, υπάρχει και στη δομή δεδομένων του πίνακα. | Σ | Λ |
| 8. Σε μια εγγραφή ενός αρχείου υπάρχουν και πεδία που περιγράφουν διάφορα χαρακτηριστικά της εγγραφής. | Σ | Λ |

- 7.
- | | | | |
|-----|---|---|---|
| 1. | Η δυαδική αναζήτηση όταν χρησιμοποιείται σε μη ταξινομημένους πίνακες είναι πιο αργή από τη σειριακή αναζήτηση. | Σ | Λ |
| 2. | Ο αλγόριθμος της σειριακής αναζήτησης χρησιμοποιείται αποκλειστικά σε ταξινομημένους πίνακες. (Ημερήσια 2006, 2011 & Εσπερινά 2011) | Σ | Λ |
| 3. | Η πιο απλή και λιγότερη αποτελεσματική μέθοδος αναζήτησης είναι η σειριακή αναζήτηση. | Σ | Λ |
| 4. | Χρησιμοποιώντας τον αλγόριθμο της σειριακής αναζήτησης μπορούμε να ψάξουμε ένα αλφαριθμητικό δεδομένο σε έναν πίνακα αλφαριθμητικών. | Σ | Λ |
| 5. | Όταν γίνεται σειριακή αναζήτηση κάποιου στοιχείου σε έναν μη ταξινομημένο πίνακα και το στοιχείο δεν υπάρχει στον πίνακα, τότε υποχρεωτικά προσπελούνται όλα τα στοιχεία. (Ημερήσια 2009) | Σ | Λ |
| 6. | Κατά την εφαρμογή της δυαδικής αναζήτησης σε έναν ταξινομημένο πίνακα προσπελούνται υποχρεωτικά όλα τα στοιχεία του πίνακα. | Σ | Λ |
| 7. | Στον πίνακα Α, που περιέχει τα ονόματα 100 μαθητών σε τυχαία σειρά, μπορούμε να χρησιμοποιήσουμε τη δυαδική αναζήτηση για να εντοπίσουμε το όνομα ενός μαθητή. | Σ | Λ |
| 8. | Η δυαδική αναζήτηση είναι πάντα ο καλύτερος τρόπος αναζήτησης. | Σ | Λ |
| 9. | Η πιο απλή μορφή αναζήτησης στοιχείου σε πίνακα είναι η σειριακή μέθοδος. (Ημερήσια 2018) | Σ | Λ |
| 10. | Όταν το στοιχείο που αναζητούμε είναι το πρώτο στον πίνακα, τότε με τη σειριακή αναζήτηση θα ελεγχθούν όλα τα στοιχεία του πίνακα. | Σ | Λ |

- 8.**
- | | | | |
|-----|--|---|---|
| 1. | Ο διδιάστατος πίνακας χρησιμοποιεί ένα μόνο δείκτη για τον προσδιορισμό των στοιχείων του. | Σ | Λ |
| 2. | Οι δείκτες ενός διδιάστατου πίνακα μπορεί να είναι και πραγματικοί αριθμοί. | Σ | Λ |
| 3. | Το στοιχείο $A[3, 4]$ βρίσκεται στην τρίτη γραμμή και στην τέταρτη στήλη του πίνακα A . | Σ | Λ |
| 4. | Σε έναν διδιάστατο πίνακα, ο αριθμός γραμμών είναι πάντα ίδιος με τον αριθμό στηλών. | Σ | Λ |
| 5. | Στην πρώτη στήλη ενός διδιάστατου πίνακα μπορούμε να αποθηκεύσουμε τα ονόματα κάποιων ατόμων και στη δεύτερη στήλη τις ηλικίες τους. | Σ | Λ |
| 6. | Σε έναν πίνακα A με 10 γραμμές και 5 στήλες, το στοιχείο $A[5, 12]$ δεν ορίζεται διότι βρίσκεται έξω από τα όρια του πίνακα. | Σ | Λ |
| 7. | Ο διδιάστατος πίνακας είναι μία δυναμική δομή δεδομένων, γιατί χρησιμοποιεί δύο δείκτες. | Σ | Λ |
| 8. | Οι πίνακες δεν μπορούν να έχουν περισσότερες από δύο διαστάσεις. (Ημερήσια 2013) | Σ | Λ |
| 9. | Σε έναν διδιάστατο πίνακα ο πρώτος δείκτης χρησιμοποιείται υποχρεωτικά για να προσδιορίζει τη γραμμή του. | Σ | Λ |
| 10. | Σε έναν διδιάστατο πίνακα υπάρχει η δυνατότητα να αποθηκεύσουμε λογικές τιμές. | Σ | Λ |
- 9.**
- | | | | |
|----|--|---|---|
| 1. | Ο μονοδιάστατος πίνακας στον οποίο θα τοποθετήσουμε το άθροισμα κάθε γραμμής ενός διδιάστατου πίνακα $A[N, M]$, πρέπει να έχει N θέσεις. | Σ | Λ |
| 2. | Το άθροισμα κάθε γραμμής ενός διδιάστατου πίνακα είναι πάντα ίσο με το άθροισμα της κάθε στήλης του. | Σ | Λ |
| 3. | Η εύρεση της θέσης του μικρότερου στοιχείου κάθε γραμμής, σημαίνει για κάθε γραμμή, τον αριθμό της στήλης όπου βρίσκεται το μικρότερο στοιχείο. | Σ | Λ |
| 4. | Η εύρεση του μέσου όρου κάθε στήλης ενός διδιάστατου πίνακα, υλοποιείται διαιρώντας το άθροισμα των στοιχείων της κάθε στήλης με το πλήθος των γραμμών του πίνακα. | Σ | Λ |
- 10.**
- | | | | |
|----|---|---|---|
| 1. | Όλες οι λειτουργίες που εφαρμόζονται σε μια δομή δεδομένων μπορούν να εφαρμοστούν σε ένα διδιάστατο πίνακα. | Σ | Λ |
| 2. | Η λειτουργία της εισαγωγής δεν μπορεί να εφαρμοστεί σε έναν διδιάστατο πίνακα. | Σ | Λ |
| 3. | Η λειτουργία της διαγραφής μπορεί να εφαρμοστεί σε έναν διδιάστατο πίνακα. | Σ | Λ |
| 4. | Η ταξινόμηση μιας στήλης δεν είναι εφικτή σε έναν διδιάστατο πίνακα αλφαριθμητικών στοιχείων. | Σ | Λ |
| 5. | Κατά την αναζήτηση ενός στοιχείου σε έναν διδιάστατο πίνακα πάντα εντοπίζουμε το στοιχείο που αναζητούμε. | Σ | Λ |

11.			
1.	Οι λειτουργίες «ώθηση» και «απώθηση» είναι οι κύριες λειτουργίες σε μία στοίβα. (Εσπερινά 2006)	Σ	Λ
2.	Κάθε στοιχείο που εισάγεται πρώτο σε μία στοίβα είναι και αυτό που εξάγεται πρώτο.	Σ	Λ
3.	Η μέθοδος επεξεργασίας LIFO περιγράφει τη διαδικασία εκείνη κατά την οποία κάθε δεδομένο που τοποθετείται τελευταίο, εξάγεται πρώτο.	Σ	Λ
4.	Η απώθηση (pop) στοιχείου γίνεται από το πίσω άκρο της στοίβας. (Επαναληπτικές Ημ. 2004)	Σ	Λ
5.	Σε μία στοίβα μπορούμε να εξαγάγουμε το τελευταίο εισαχθέν στοιχείο, χωρίς να έχει εξαχθεί το προτελευταίο στοιχείο.	Σ	Λ
6.	Υποχειλίσση σε μια στοίβα συμβαίνει, όταν η στοίβα είναι κενή και εκτελέσουμε τη λειτουργία της απώθησης.	Σ	Λ
7.	Μία στοίβα μπορούμε να την υλοποιήσουμε με τη βοήθεια ενός μονοδιάστατου πίνακα.	Σ	Λ
8.	Όταν εισάγουμε ένα στοιχείο σε μία στοίβα, πρώτα αυξάνεται ο δείκτης που δείχνει την κορυφή της και στη συνέχεια εισάγεται το στοιχείο.	Σ	Λ
9.	Εκτελώντας τη λειτουργία της εξαγωγής, αφαιρούμε το στοιχείο που βρίσκεται στην κορυφή μιας στοίβας.	Σ	Λ
10.	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια άδεια στοίβα.	Σ	Λ
11.	Στη στοίβα το στοιχείο που μπαίνει πρώτο βγαίνει πάντα πρώτο.	Σ	Λ
12.	Σε μια γεμάτη στοίβα, εκτός από την απώθηση μπορεί να εφαρμοστεί και η λειτουργία της ώθησης.	Σ	Λ
13.	Ο «εμπρός» και ο «πίσω», είναι δύο δείκτες που απαιτούνται για τη διαχείριση μιας στοίβας.	Σ	Λ
14.	Η λειτουργία της ώθησης υλοποιείται πάντα στη κορυφή μιας στοίβας.	Σ	Λ
15.	Το φαινόμενο της υπερχειλίσσης παρουσιάζεται όταν προσπαθήσουμε να προσθέσουμε ένα στοιχείο σε μία ήδη γεμάτη στοίβα.	Σ	Λ
16.	Κατά τη διαδικασία της ώθησης πρέπει να ελέγχεται αν η στοίβα είναι γεμάτη. (Επαναληπτικές Ημ. 2004)	Σ	Λ
17.	Η ώθηση (push) στοιχείου είναι μία από τις λειτουργίες της ουράς. (Επαναληπτικές Ημ. 2004)	Σ	Λ
18.	Η λειτουργία της ώθησης σχετίζεται με τη δομή της στοίβας. (Επαναληπτικές Ημ. 2009)	Σ	Λ
19.	Στη στοίβα, ο ίδιος δείκτης μάς δίνει, τόσο τη θέση του στοιχείου που μπορεί να εξαχθεί, όσο και τη θέση εκείνου που εισήλθε τελευταίο. (Επαναληπτικές Ημ. 2016)	Σ	Λ

12.

- | | | |
|--|---|---|
| 1. Στην ουρά, κάθε στοιχείο της εξάγεται από το μπροστινό άκρο της. | Σ | Λ |
| 2. Η μέθοδος επεξεργασίας «πρώτο μέσα πρώτο έξω» (FIFO) εφαρμόζεται στη δομή δεδομένων ΟΥΡΑ. (Ημερήσια 2012 & Εσπερινά 2012) | Σ | Λ |
| 3. Στην ουρά, με τη λειτουργία της εισαγωγής μπορούν να εισαχθούν περισσότερα από ένα δεδομένα μαζί. | Σ | Λ |
| 4. Για να διαχειριστούμε μία ουρά, πρέπει να χρησιμοποιήσου-με δύο δείκτες: έναν που δείχνει το πίσω άκρο της ουράς και έναν που δείχνει το μέσον της ουράς. | Σ | Λ |
| 5. Κατά την εξαγωγή ενός στοιχείου στην ουρά, αυξάνεται ο δείκτης που δείχνει το πίσω άκρο της. | Σ | Λ |
| 6. Η ουρά και η στοίβα μπορούν να υλοποιηθούν με δομή πίνακα. (Επαναληπτικές Ημ. 2004) | Σ | Λ |
| 7. Όταν πραγματοποιείται η εισαγωγή ενός στοιχείου σε μια ουρά, αυτό τοποθετείται στο μπροστινό άκρο της. | Σ | Λ |
| 8. Η ώθηση (push) στοιχείου είναι μία από τις λειτουργίες της ουράς. (Επαναληπτικές Ημ. 2004) | Σ | Λ |
| 9. Στην ουρά το στοιχείο που μπαίνει πρώτο βγαίνει και πρώτο. | Σ | Λ |
| 10. Η λειτουργία της εξαγωγής μπορεί να εκτελεστεί σε μια γεμάτη ουρά. | Σ | Λ |
| 11. Η υλοποίηση της ουράς χρησιμοποιεί μία μεταβλητή-δείκτη για την εκτέλεση των δύο βασικών λειτουργιών της. | Σ | Λ |
| 12. Οι λειτουργίες εισαγωγή και εξαγωγή είναι οι κύριες λειτουργίες σε μία ουρά. | Σ | Λ |
| 13. Όταν εισάγουμε ένα στοιχείο σε μία ουρά, πρώτα αυξάνεται ο δείκτης που δείχνει την κορυφή της και στη συνέχεια εισάγεται το στοιχείο. | Σ | Λ |
| 14. Εκτελώντας τη λειτουργία της εξαγωγής, αφαιρούμε το στοιχείο που βρίσκεται στην κορυφή μιας στοίβας. | Σ | Λ |
| 15. Σε μια άδεια ουρά, εκτός από την εισαγωγή μπορεί να εφαρμοστεί και η λειτουργία της εξαγωγής. | Σ | Λ |
| 16. Ο «εμπρός» και ο «πίσω», είναι δύο δείκτες που απαιτούνται για τη διαχείριση μιας ουράς. | Σ | Λ |
| 17. Στην ουρά A[50], όταν οι δείκτες «εμπρός» και «πίσω», έχουν ως τιμή έναν θετικό αριθμό μικρότερο του 50 η ουρά έχει πάντα μέσα της ένα στοιχείο. | Σ | Λ |
| 18. Όταν οι δείκτες «εμπρός» και «πίσω», έχουν την τιμή μηδέν η ουρά είναι άδεια. | Σ | Λ |
| 19. Στον πίνακα A[50] που υλοποιεί μια ουρά, αν οι δείκτης «εμπρός» και «πίσω», έχουν τιμές 1 και 50, τότε η ουρά είναι γεμάτη. | Σ | Λ |

**ΠΑΡΑΡΤΗΜΑ ΑΛΓΟΡΙΘΜΩΝ
ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ**



ΒΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΜΟΝΟΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ

- **ΓΕΜΙΣΜΑ ΠΙΝΑΚΑ Π[N].**

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
ΔΙΑΒΑΣΕ Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

- **ΕΜΦΑΝΙΣΗ ΠΙΝΑΚΑ Π[N].**

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
ΓΡΑΨΕ Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

- **ΕΥΡΕΣΗ ΑΘΡΟΙΣΜΑΤΟΣ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N].**

Σ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
Σ←Σ+Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Σ

- **ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N].**

ΠΛ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
ΑΝ Π[I] (ΣΥΝΘΗΚΗ) ΤΟΤΕ
ΠΛ←ΠΛ+1
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ

- **ΕΥΡΕΣΗ ΜΕΣΟΥ ΟΡΟΥ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N].**

Σ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
Σ←Σ+Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ←Σ/N
ΓΡΑΨΕ ΜΟ

- **ΕΥΡΕΣΗ ΓΙΝΟΜΕΝΟΥ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N].**

Γ←1
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
Γ←Γ*Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Γ

- **ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ ΣΕ ΠΙΝΑΚΑ Π[N].**

ΜΑΧ←Π[1]
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
ΑΝ Π[I]>ΜΑΧ ΤΟΤΕ
ΜΑΧ←Π[I]
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ

Αν θέλουμε να γίνονται
οι ενέργειες με βάση
κάποια συγκεκριμένα
κριτήρια βάζουμε
ΑΝ <κριτήριο> ΤΟΤΕ

• **ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΟΥ ΣΕ ΠΙΝΑΚΑ Π[N].**

```
MIN ← Π[1]
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  ΑΝ Π[I] < MIN ΤΟΤΕ
    MIN ← Π[I]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ MIN
```

• **ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ ΚΑΙ ΘΕΣΗΣ ΣΕ ΠΙΝΑΚΑ Π[N].**

```
MAX ← Π[1]
MAXΘΕΣΗ ← 1
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  ΑΝ Π[I] > MAX ΤΟΤΕ
    MAX ← Π[I]
    MAXΘΕΣΗ ← I
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ MAX, MAXΘΕΣΗ
```

• **ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΟΥ ΚΑΙ ΘΕΣΗΣ ΣΕ ΠΙΝΑΚΑ Π[N].**

```
MIN ← Π[1]
MINΘΕΣΗ ← 1
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  ΑΝ Π[I] < MIN ΤΟΤΕ
    MIN ← Π[I]
    MINΘΕΣΗ ← I
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ MIN, MINΘΕΣΗ
```

• **ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ ΚΑΙ ΘΕΣΕΩΝ ΣΕ ΠΙΝΑΚΑ Π[N].**

```
ΜΕΓ ← Π[1]
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  ΑΝ Π[I] > ΜΕΓ ΤΟΤΕ
    ΜΕΓ ← Π[I]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΑΝ Π[I] = ΜΕΓ ΤΟΤΕ
    ΓΡΑΨΕ ΜΕΓ, I
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• **ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΟΥ ΚΑΙ ΘΕΣΕΩΝ ΣΕ ΠΙΝΑΚΑ Π[N].**

```
ΕΛ ← Π[1]
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  ΑΝ Π[I] < ΕΛ ΤΟΤΕ
    ΕΛ ← Π[I]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΑΝ Π[I] = ΕΛ ΤΟΤΕ
    ΓΡΑΨΕ ΕΛ, I
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• **ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ ΜΕΓΙΣΤΩΝ (ΕΛΑΧΙΣΤΩΝ) ΣΕ ΠΙΝΑΚΑ Π[N].**

```

ΜΕΓ ← Π[1]
ΠΛ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΑΝ Π[Ι] > ΜΕΓ ΤΟΤΕ
    ΜΕΓ ← Π[Ι]
    ΠΛ ← 1
  ΑΛΛΙΩΣ_ΑΝ Π[Ι] = ΜΕΓ ΤΟΤΕ
    ΠΛ ← ΠΛ + 1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΕΓ, ΠΛ
  
```

• **ΕΥΡΕΣΗ ΔΥΟ ΜΕΓΙΣΤΩΝ (ΕΛΑΧΙΣΤΩΝ) ΣΕ ΠΙΝΑΚΑ Π[N].**

```

ΜΕΓ1 ← Π[1]
ΜΕΓ2 ← Π[1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΑΝ Π[Ι] > ΜΕΓ1 ΤΟΤΕ
    ΜΕΓ2 ← ΜΕΓ1
    ΜΕΓ1 ← Π[Ι]
  ΑΛΛΙΩΣ_ΑΝ Π[Ι] > ΜΕΓ2 ΤΟΤΕ
    ΜΕΓ2 ← Π[Ι]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΕΓ1, ΜΕΓ2
  
```

• **ΠΑΡΑΛΛΗΛΟΙ ΠΙΝΑΚΕΣ Π[N] – Α[N].**

Π[N]				
ΜΑΡΙΑ	ΓΙΑΝΝΗΣ	ΚΩΣΤΑΣ	ΔΗΜΗΤΡΑ	ΕΛΕΝΗ
(1)	(2)	(3)	(4)	(5)
↑↓	↑↓	↑↓	↑↓	↑↓
Α[N]				
18	15	10	16	17
(1)	(2)	(3)	(4)	(5)

ΤΑ ΣΤΟΙΧΕΙΑ ΤΩΝ ΠΙΝΑΚΩΝ ΠΟΥ ΕΧΟΥΝ ΤΟΝ ΙΔΙΟ ΔΕΙΚΤΗ ΣΧΕΤΙΖΟΝΤΑΙ ΜΕΤΑΞΥ ΤΟΥΣ

• **ΠΙΝΑΚΑΣ ΜΕΤΡΗΤΩΝ**

```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΜΕΤΡ[Ι] ← 0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  
```

```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΜΕΤΡ[Ι] ← 0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  
```

```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Μ
  ΑΝ Π[Ι] ΣΥΝΘΗΚΗ 1 ΤΟΤΕ
    ΜΕΤΡ[1] ← ΜΕΤΡ[1] + 1
  ΑΛΛΙΩΣ_ΑΝ Π[Ι] ΣΥΝΘΗΚΗ 2 ΤΟΤΕ
    ΜΕΤΡ[2] ← ΜΕΤΡ[2] + 1
  .....
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  
```

```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Μ
  ΑΝ Π[Ι] ΣΥΝΘΗΚΗ ΤΟΤΕ
    ΔΕΙΚΤΗΣ ← Π[Ι]
    Π[ΔΕΙΚΤΗΣ] ← Π[ΔΕΙΚΤΗΣ] + 1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  
```

• **ΣΥΓΧΩΝΕΥΣΗ ΠΙΝΑΚΩΝ**

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΔΙΑΒΑΣΕ A[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ M
  ΔΙΑΒΑΣΕ B[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΝΠ[I] ← A[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Κ ← 0
ΓΙΑ I ΑΠΟ N+1 ΜΕΧΡΙ M
  Κ ← Κ+1
  ΝΠ[I] ← B[Κ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
I ← 1
J ← 1
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ N+M
  ΑΝ I ≤ N ΚΑΙ J ≤ M ΤΟΤΕ
    ΑΝ A[I] < B[J] ΤΟΤΕ
      Γ[Κ] ← A[I]
      I ← I+1
    ΑΛΛΙΩΣ
      Γ[Κ] ← B[J]
      J ← J + 1
    ΤΕΛΟΣ_ΑΝ
  ΑΛΛΙΩΣ_ΑΝ I ≤ N ΤΟΤΕ
    Γ[Κ] ← A[I]
    I ← I+1
  ΑΛΛΙΩΣ
    Γ[Κ] ← B[J]
    J ← J + 1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΣΤΟΙΧΕΙΑ ΠΙΝΑΚΑ

ΤΟ ΣΤΟΙΧΕΙΟ A[I] ΕΧΕΙ ΠΡΟΗΓΟΥΜΕΝΟ ΤΟ A[I-1] ΚΑΙ ΕΠΟΜΕΝΟ ΤΟ A[I+1]

ΟΡΙΑ ΠΙΝΑΚΩΝ

ΕΛΕΓΧΟΣ	ΟΡΙΟ	Σ
ΟΛΑ ΤΑ ΣΤΟΙΧΕΙΑ	N	A[I]
ΖΕΥΓΗ ΔΙΑΔΟΧΙΚΩΝ ΣΤΟΙΧΕΙΩΝ	N-1	A[I], A[I+1]
ΤΡΙΑΔΕΣ ΔΙΑΔΟΧΙΚΩΝ ΣΤΟΙΧΕΙΩΝ	N-2	A[I], A[I+1], A[I+2]

ΑΝΑΖΗΤΗΣΗ

Αναζήτηση χρησιμοποιούμε όταν από την εκφώνηση ζητείται:

- να διαβάσουμε ένα δεδομένο και για αυτό να βρούμε κάποια στοιχεία που υπάρχουν σε έναν ή περισσότερους πίνακες (διαβάζουμε ΑΤ και να εμφανίσουμε όνομα)
- ζητείται για ένα δεδομένο να βρούμε κάποια στοιχεία που υπάρχουν σε ένα ή περισσότερους πίνακες τα οποία σχετίζονται με αυτό το δεδομένο (να βρούμε ο μαθητής Παπαδόπουλος ποιο μέσο όρο έχει)

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΜΗ ΤΑΞΙΝΟΜΗΜΕΝΟ ΠΙΝΑΚΑ Π[Ν] (ΣΤΟ ΠΡΩΤΟ ΠΟΥ ΒΡΙΣΚΕΙ ΣΤΑΜΑΤΑ)

```
ΔΙΑΒΑΣΕ ΖΗΤ_ΤΙΜΗ
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
I ← 1
ΘΕΣΗ ← 0
'ΟΣΟ (ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ) ΚΑΙ (I<=N) ΕΠΑΝΑΛΑΒΕ
  ΑΝ Π[I]=ΖΗΤ_ΤΙΜΗ ΤΟΤΕ
    ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
    ΘΕΣΗ ← I
  ΑΛΛΙΩΣ
    I ← I+1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ ΒΡΕΘΗΚΕ=ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ ΣΤΗ ΘΕΣΗ', ΘΕΣΗ
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ ΣΤΟΝ ΠΙΝΑΚΑ'
ΤΕΛΟΣ_ΑΝ
```

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΜΗ ΤΑΞΙΝΟΜΗΜΕΝΟ ΠΙΝΑΚΑ Π[Ν] (ΝΑ ΒΡΕΙ ΤΟ ΤΕΛΕΥΤΑΙΟ)

```
ΔΙΑΒΑΣΕ ΖΗΤ_ΤΙΜΗ
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΑΝ Π[I]=ΖΗΤ_ΤΙΜΗ ΤΟΤΕ
    ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ
    ΘΕΣΗ←I
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ ΣΤΟΝ ΠΙΝΑΚΑ'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ ΣΤΗ ΘΕΣΗ', ΘΕΣΗ
ΤΕΛΟΣ_ΑΝ
```

**ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΜΗ ΤΑΞΙΝΟΜΗΜΕΝΟ ΠΙΝΑΚΑ Π[Ν]
(ΝΑ ΤΑ ΒΡΕΙ ΟΛΑ)**

```
ΔΙΑΒΑΣΕ ΖΗΤ_ΤΙΜΗ
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΑΝ Π[Ι]=ΖΗΤ_ΤΙΜΗ ΤΟΤΕ
    ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ
    ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ ΣΤΗ ΘΕΣΗ', Ι
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ ΣΤΟΝ ΠΙΝΑΚΑ'
ΤΕΛΟΣ_ΑΝ
```

**ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΜΗ ΤΑΞΙΝΟΜΗΜΕΝΟ ΠΙΝΑΚΑ Π[Ν]
(ΝΑ ΤΑ ΒΡΕΙ ΟΛΑ – ΝΑ ΤΑ ΑΠΟΘΗΚΕΥΕΙ ΣΕ ΝΕΟ ΠΙΝΑΚΑ)**

```
ΔΙΑΒΑΣΕ ΖΗΤ_ΤΙΜΗ
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
Κ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΑΝ Π[Ι]=ΖΗΤ_ΤΙΜΗ ΤΟΤΕ
    Κ←Κ+1
    ΝΠ[Κ]← Ι
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ Κ<>0 ΤΟΤΕ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Κ
    ΓΡΑΨΕ ΝΠ[Ι]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
```

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΤΑΞΙΝΟΜΗΜΕΝΟ ΠΙΝΑΚΑ (ΑΥΞΟΥΣΑ)

```
DONE ← ΨΕΥΔΗΣ
POSITION ← 0
I ← 1
'ΟΣΟ (DONE = ΨΕΥΔΗΣ) ΚΑΙ (I <= N) ΕΠΑΝΑΛΑΒΕ
  ΑΝ (TABLE[I] > KEY) ΤΟΤΕ
    DONE ← ΑΛΗΘΗΣ
  ΑΛΛΙΩΣ_ΑΝ (TABLE[I] = KEY) ΤΟΤΕ
    DONE ← ΑΛΗΘΗΣ
    POSITION ← I
  ΑΛΛΙΩΣ
    I ← I + 1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ (POSITION <> 0) ΤΟΤΕ
  ΓΡΑΨΕ 'ΤΟ ΣΤΟΙΧΕΙΟ, KEY, 'ΒΡΕΘΗΚΕ ΣΤΗ ΘΕΣΗ', POSITION
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΤΟ ΣΤΟΙΧΕΙΟ', KEY, 'ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
```

***** 'Όταν ο πίνακας είναι ταξινομημένος κατά **φθίνουσα** σειρά αλλάζει η φορά σύγκρισης στο Αν και γίνεται **ΑΝ (TABLE[I] < KEY) ΤΟΤΕ *******

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΤΑΞΙΝΟΜΗΜΕΝΟ ΠΙΝΑΚΑ (ΑΥΞΟΥΣΑ)

```
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
ΔΕΝΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
I ← 1
'ΟΣΟ (ΔΕΝΒΡΕΘΗΚΕ = ΑΛΗΘΗΣ) ΚΑΙ (I ≤ N) ΕΠΑΝΑΛΑΒΕ
  ΑΝ (TABLE[I] = KEY) ΤΟΤΕ
    ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ', I
    ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
  ΑΛΛΙΩΣ_ΑΝ (TABLE[I] > KEY) ΤΟΤΕ
    ΔΕΝΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
  ΤΕΛΟΣ_ΑΝ
  I ← I + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ (ΥΠΑΡΧΕΙ ΜΙΑ ΦΟΡΑ)

```
i ← 1
ΟΣΟ A[i] <> key ΚΑΙ i ≤ 100 ΕΠΑΝΑΛΑΒΕ
  i ← i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ A[i] = key ΤΟΤΕ
  ΓΡΑΨΕ 'Βρέθηκε στη θέση:', i
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Δεν βρέθηκε'
ΤΕΛΟΣ_ΑΝ
```

ΔΥΑΔΙΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΠΙΝΑΚΑ (ΤΑΞΙΝΟΜΗΜΕΝΟΣ ΠΙΝΑΚΑΣ – ΑΥΞΟΥΣΑ)

```
LEFT ← 1
RIGHT ← N
POS ← 0
FLAG ← ΨΕΥΔΗΣ
'ΟΣΟ (LEFT<=RIGHT) ΚΑΙ (FLAG = ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ
    M ← (LEFT+RIGHT) DIV 2
    ΑΝ A[M]=KEY ΤΟΤΕ
        POS ← M
        FLAG ← ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ_ΑΝ A[M]< KEY ΤΟΤΕ
        LEFT ← M+1
    ΑΛΛΙΩΣ
        RIGHT← M-1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ FLAG = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'ΤΟ ΣΤΟΙΧΕΙΟ,', KEY , 'ΥΠΑΡΧΕΙ ΣΤΗ ΘΕΣΗ:', POS
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΤΟ ΣΤΟΙΧΕΙΟ,', KEY , 'ΔΕΝ ΥΠΑΡΧΕΙ ΣΤΟΝ ΠΙΝΑΚΑ'
ΤΕΛΟΣ_ΑΝ
```

***** Όταν ο πίνακας είναι ταξινομημένος κατά **φθίνουσα** σειρά αλλάζει η φορά σύγκρισης στο αλλιώς_αν και γίνεται **ΑΛΛΙΩΣ_ΑΝ A[M]> KEY ΤΟΤΕ*******

ΑΥΞΟΥΣΑ

```
ΑΡΧΗ ← 1
ΤΕΛΟΣ ← 100
ΜΕΣΗ ← (ΑΡΧΗ + ΤΕΛΟΣ) DIV 2
ΟΣΟ ΑΡΧΗ <=ΤΕΛΟΣ ΚΑΙ A[ΜΕΣΗ]<> KEY ΕΠΑΝΑΛΑΒΕ
    ΑΝ KEY < A[ΜΕΣΗ] ΤΟΤΕ
        ΤΕΛΟΣ ← ΜΕΣΗ-1
    ΑΛΛΙΩΣ
        ΑΡΧΗ ← ΜΕΣΗ + 1
    ΤΕΛΟΣ_ΑΝ
    ΜΕΣΗ ← (ΑΡΧΗ + ΤΕΛΟΣ) DIV 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ A[ΜΕΣΗ] = KEY ΤΟΤΕ
    ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ ΣΤΗ ΘΕΣΗ ',ΜΕΣΗ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
```

ΦΘΙΝΟΥΣΑ

```
ΑΡΧΗ ← 1
ΤΕΛΟΣ ← 100
ΜΕΣΗ ← (ΑΡΧΗ + ΤΕΛΟΣ) DIV 2
ΟΣΟ ΑΡΧΗ<=ΤΕΛΟΣ ΚΑΙ A[ΜΕΣΗ]<>KEY ΕΠΑΝΑΛΑΒΕ
    ΑΝ KEY < A[ΜΕΣΗ] ΤΟΤΕ
        ΑΡΧΗ ← ΜΕΣΗ + 1
    ΑΛΛΙΩΣ
        ΤΕΛΟΣ ← ΜΕΣΗ - 1
    ΤΕΛΟΣ_ΑΝ
    ΜΕΣΗ ← (ΑΡΧΗ + ΤΕΛΟΣ) DIV 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ A[ΜΕΣΗ] = KEY ΤΟΤΕ
    ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ ΣΤΗ ΘΕΣΗ ',ΜΕΣΗ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
```

```

ΔΙΑΒΑΣΕ ΚΕΥ
αρχ <- 1
τέλος <- 100
δεικτης <- ΨΕΥΔΗΣ
ΟΣΟ αρχ <= τέλος ΚΑΙ δεικτης = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
  μέση <- (αρχ + τέλος) div 2
  ΑΝ μέση >= 1 ΚΑΙ μέση <= 100 ΤΟΤΕ
    ΑΝ ΚΕΥ < Α[μέση] ΤΟΤΕ
      τέλος <- μέση - 1
    ΑΛΛΙΩΣ_ΑΝ ΚΕΥ > Α[μέση] ΤΟΤΕ
      αρχ <- μέση + 1
    ΑΛΛΙΩΣ
      δεικτης <- ΑΛΗΘΗΣ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ δεικτης = ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Δε βρέθηκε'
ΤΕΛΟΣ_ΑΝ

```

```

ΔΙΑΒΑΣΕ ΚΕΥ
αρχ <- 1
τέλος <- 5
δεικτης <- ΨΕΥΔΗΣ
ΟΣΟ αρχ <= τέλος ΚΑΙ δεικτης = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
  μέση <- (αρχ + τέλος) div 2
  ΑΝ μέση >= 1 ΚΑΙ μέση <= 5 ΤΟΤΕ
    ΑΝ ΚΕΥ < Α[μέση] ΤΟΤΕ
      αρχ <- μέση + 1
    ΑΛΛΙΩΣ_ΑΝ ΚΕΥ > Α[μέση] ΤΟΤΕ
      τέλος <- μέση - 1
    ΑΛΛΙΩΣ
      δεικτης <- ΑΛΗΘΗΣ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ δεικτης = ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Δε βρέθηκε'
ΤΕΛΟΣ_ΑΝ

```

**ΔΥΑΔΙΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΠΙΝΑΚΑ (ΤΑΞΙΝΟΜΗΜΕΝΟΣ ΠΙΝΑΚΑΣ – ΑΥΞΟΥΣΑ)
(ΤΟ ΣΤΟΙΧΕΙΟ ΥΠΑΡΧΕΙ ΠΟΛΛΕΣ ΦΟΡΕΣ)**

```
LEFT ← 1
RIGHT ← N
POS ← 0
FLAG ← ΨΕΥΔΗΣ
'ΟΣΟ (LEFT<=RIGHT) ΚΑΙ (FLAG = ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ
    M ← (LEFT+RIGHT) DIV 2
    AN A[M]=KEY ΤΟΤΕ
        POS ← M
        FLAG ← ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ_ΑΝ A[M]< KEY ΤΟΤΕ
        LEFT ← M+1
    ΑΛΛΙΩΣ
        RIGHT← M-1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ FLAG = ΑΛΗΘΗΣ ΤΟΤΕ
    I←1
    ΠΛ←1
    ΓΡΑΨΕ POS
    ΟΣΟ A[POS+I] = KEY ΕΠΑΝΑΛΑΒΕ           !ΕΛΕΓΧΟΣ ΑΝ ΥΠΑΡΧΕΙ ΜΕΤΑ
        ΠΛ←ΠΛ+1
        ΓΡΑΨΕ POS+I
        I←I+1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    I←1
    ΟΣΟ A[POS-I] = KEY ΕΠΑΝΑΛΑΒΕ           !ΕΛΕΓΧΟΣ ΑΝ ΥΠΑΡΧΕΙ ΠΡΙΝ
        ΠΛ←ΠΛ+1
        ΓΡΑΨΕ POS-I
        I←I+1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ ΠΛ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΤΟ ΣΤΟΙΧΕΙΟ ΔΕΝ ΥΠΑΡΧΕΙ ΣΤΟΝ ΠΙΝΑΚΑ'
ΤΕΛΟΣ_ΑΝ
```

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ Π[N,M]
(ΥΠΑΡΧΕΙ ΜΙΑ ΦΟΡΑ)

```
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
I ← 1
ΓΡΑΜΜΗ ← 0
ΣΤΗΛΗ ← 0
'ΟΣΟ I <= N ΚΑΙ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
  J ← 1
  'ΟΣΟ J <= M ΚΑΙ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ Π[I,J] = ΖΗΤ.ΤΙΜΗ ΤΟΤΕ
      ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
      ΓΡΑΜΜΗ ← I
      ΣΤΗΛΗ ← J
    ΑΛΛΙΩΣ_ΑΝ
      J ← J + 1
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  I ← I + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ ΒΡΕΘΗΚΕ = ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'Η ΤΙΜΗ ΒΡΕΘΗΚΕ ΣΤΗ ΓΡΑΜΜΗ', ΓΡΑΜΜΗ, 'ΚΑΙ ΣΤΗ ΣΤΗΛΗ', ΣΤΗΛΗ
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Η ΤΙΜΗ ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
```

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ Π[N,M]
(ΥΠΑΡΧΕΙ ΠΟΛΛΕΣ ΦΟΡΕΣ – ΕΜΦΑΝΙΖΕΙ ΤΕΛΕΥΤΑΙΟ)

```
ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
    ΑΝ Π[I,J] = ΖΗΤ.ΤΙΜΗ ΤΟΤΕ
      ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ ΒΡΕΘΗΚΕ = ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'Η ΤΙΜΗ ΒΡΕΘΗΚΕ', I,J
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Η ΤΙΜΗ ΔΕΝ ΒΡΕΘΗΚΕ'
ΤΕΛΟΣ_ΑΝ
```

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ Π[N,M]
(ΥΠΑΡΧΕΙ ΠΟΛΛΕΣ ΦΟΡΕΣ – ΕΜΦΑΝΙΖΕΙ ΟΛΑ)

```
ΠΛ ← 0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
    ΑΝ Π[I,J] = ΖΗΤ.ΤΙΜΗ ΤΟΤΕ
      ΓΡΑΨΕ 'ΒΡΕΘΗΚΕ ΣΤΗΝ ΓΡΑΜΜΗ', I, 'ΚΑΙ ΣΤΗΝ ΣΤΗΛΗ', J
      ΠΛ ← ΠΛ + 1
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ ΠΛ <> 0 ΤΟΤΕ
  ΓΡΑΨΕ 'Η ΤΙΜΗ ΒΡΕΘΗΚΕ', ΠΛ, 'ΦΟΡΕΣ'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Η ΤΙΜΗ ΔΕΝ ΒΡΕΘΗΚΕ'
```


ΤΑΞΙΝΟΜΗΣΗ

ΤΑΞΙΝΟΜΗΣΗ ΕΥΘΕΙΑΣ ΑΝΤΑΛΛΑΓΗΣ-ΦΥΣΑΛΙΔΑ (ΑΥΞΟΥΣΑ) ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N]

```
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν
ΓΙΑ Κ ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
  ΑΝ Π[Κ-1]>Π[Κ] ΤΟΤΕ          ΑΝΤΙΜΕΤΕΘΕΣΕ (Π[Κ-1],Π[Κ])
    ΤΕΜΡ←Π[Κ-1]
    Π[Κ-1]←Π[Κ]
    Π[Κ]←ΤΕΜΡ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΑΞΙΝΟΜΗΣΗ ΕΥΘΕΙΑΣ ΑΝΤΑΛΛΑΓΗΣ-ΦΥΣΑΛΙΔΑ (ΦΘΙΝΟΥΣΑ) ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N]

```
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν
ΓΙΑ Κ ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
  ΑΝ Π[Κ-1]<Π[Κ] ΤΟΤΕ          ΑΝΤΙΜΕΤΕΘΕΣΕ (Π[Κ-1],Π[Κ])
    ΤΕΜΡ←Π[Κ-1]
    Π[Κ-1]←Π[Κ]
    Π[Κ]←ΤΕΜΡ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

******* Όταν έχουμε περισσότερους από έναν πίνακα που συσχετίζονται τότε αντιμεταθέτουμε και τα στοιχεία των άλλων πινάκων*******

**** Περίπτωση ίσων στοιχείων του πίνακα --- ταξινόμηση με στοιχεία άλλου πίνακα****

```
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν
  ΓΙΑ J ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ ΒΗΜΑ -1
    ΑΝ Μ[J-1]< Μ[J] ΤΟΤΕ
      ΤΕΜΡ1 ← Μ[J-1]
      Μ[J-1] ← Μ[J]
      Μ[J] ← ΤΕΜΡ1
      ΤΕΜΡ2 ← Τ[J-1]
      Τ[J-1] ← Τ[J]
      Τ[J] ← ΤΕΜΡ2
      ΤΕΜΡ3 ← Ε[J-1]
      Ε[J-1] ← Ε[J]
      Ε[J] ← ΤΕΜΡ3
    ΑΛΛΙΩΣ ΑΝ Μ[J-1] = Μ[J] ΤΟΤΕ
      ΑΝ Ε[J-1] > Ε[J] ΤΟΤΕ
        ΤΕΜΡ1 ← Ε[J-1]
        Ε[J-1] ← Ε[J]
        Ε[J] ← ΤΕΜΡ1
        ΤΕΜΡ2 ← Τ[J-1]
        Τ[J-1] ← Τ[J]
        Τ[J] ← ΤΕΜΡ2
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΑΞΙΝΟΜΗΣΗ ΕΥΘΕΙΑΣ ΑΝΤΑΛΛΑΓΗΣ (ΑΥΞΟΥΣΑ) ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ - ο πίνακας είναι ταξινομημένος, τότε σταματά ο έλεγχος των στοιχείων για ταξινόμηση.

```
Flag ← ΑΛΗΘΗΣ
i ← 2
ΟΣΟ i <= 100 ΚΑΙ flag = ΑΛΗΘΗΣ ΕΠΑΝΑΛΑΒΕ
    flag ← ΨΕΥΔΗΣ
    ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
        ΑΝ A[j-1] > A[j] ΤΟΤΕ
            Κ ← A[j - 1]
            A[j-1] ← A[j]
            A[j] ← Κ
            flag ← ΑΛΗΘΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    i ← i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
I←2
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΒΡΕΘΗΚΕ← ΑΛΗΘΗΣ
    ΓΙΑ J ΑΠΟ N ΜΕΧΡΙ I ΜΕ_ΒΗΜΑ -1
        ΑΝ Π[J-1]>Π[J] ΤΟΤΕ
            TEMP←Π[J-1]
            Π[J-1]←Π[J]
            Π[J]←TEMP
            ΒΡΕΘΗΚΕ← ΨΕΥΔΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
I←I+1
ΜΕΧΡΙΣ_ΟΤΟΥ I>N Η ΒΡΕΘΗΚΕ= ΑΛΗΘΗΣ
```

```
I←2
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΒΡΕΘΗΚΕ← ΨΕΥΔΗΣ
    ΓΙΑ J ΑΠΟ N ΜΕΧΡΙ I ΜΕ_ΒΗΜΑ -1
        ΑΝ Π[J-1]>Π[J] ΤΟΤΕ
            TEMP←Π[J-1]
            Π[J-1]←Π[J]
            Π[J]←TEMP
            ΒΡΕΘΗΚΕ← ΑΛΗΘΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
I←I+1
ΜΕΧΡΙΣ_ΟΤΟΥ I>N Η ΒΡΕΘΗΚΕ= ΨΕΥΔΗΣ
```

ΠΑΡΑΛΛΑΓΕΣ ΕΥΘΕΙΑΣ ΑΝΤΑΛΛΑΓΗΣ-ΦΥΣΑΛΙΔΑ

ΕΝΑΛΛΑΞ ΑΡΧΗ ΤΕΛΟΣ

```
ΑΡΧΗ←1
ΤΕΛΟΣ←N-1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ
    ΓΙΑ I ΑΠΟ ΑΡΧΗ ΜΕΧΡΙ ΤΕΛΟΣ
        ΑΝ ΠI]>Π[I+1] ΤΟΤΕ
            TEMP←Π[I+1]
            Π[I+1]← Π[I]
            Π[I]←TEMP
            ΒΡΕΘΗΚΕ← ΑΛΗΘΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ←ΤΕΛΟΣ-1
ΓΙΑ I ΑΠΟ ΑΡΧΗ ΜΕΧΡΙ ΤΕΛΟΣ ΜΕ_ΒΗΜΑ -1
    ΑΝ ΠI]>Π[I+1] ΤΟΤΕ
        TEMP←Π[I+1]
        Π[I+1]← Π[I]
        Π[I]←TEMP
        ΒΡΕΘΗΚΕ← ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΡΧΗ←ΑΡΧΗ+1
```


ΜΕΧΡΙΣ_ΟΤΟΥ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ

ΑΡΤΙΟΙ ΠΕΡΙΤΤΟΙ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ N-1 ΜΕ_ΒΗΜΑ 2

ΑΝ Π[Ι]>Π[Ι+1] ΤΟΤΕ

TEMP←Π[Ι+1]

Π[Ι+1]←Π[Ι]

Π[Ι]←TEMP

ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ N-1 ΜΕ_ΒΗΜΑ 2

ΑΝ Π[Ι]>Π[Ι+1] ΤΟΤΕ

TEMP←Π[Ι+1]

Π[Ι+1]←Π[Ι]

Π[Ι]←TEMP

ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΕΧΡΙΣ_ΟΤΟΥ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (ΑΥΞΟΥΣΑ) Π[Ν]

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N

K ← I

ΓΙΑ J ΑΠΟ I+1 ΜΕΧΡΙ N

ΑΝ Π[J-1] < Π[K] ΤΟΤΕ

K ← J

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

TEMP ← Π[I]

Π[I] ← Π[K]

Π[K] ← TEMP

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

*****Σε φθίνουσα ταξινόμηση αλλάζει φορά η ανισότητα στο Αν Π[J-1] > Π[K] *****

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N

MIN ← Π[I]

Θ ← I

ΓΙΑ J ΑΠΟ I+1 ΜΕΧΡΙ N

ΑΝ Π[J] < MIN ΤΟΤΕ

MIN ← Π[J]

Θ ← J

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Π[Θ] ← Π[I]

Π[I] ← MIN

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N - 1

K ← I

X ← TABLE[I]

ΓΙΑ J ΑΠΟ I + 1 ΜΕΧΡΙ N

ΑΝ X > TABLE[J] ΤΟΤΕ

K ← J

X ← TABLE[J]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

TABLE[K] ← TABLE[I]

TABLE[I] ← X

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΑΞΙΝΟΜΗΣΗ ΕΥΘΕΙΑΣ ΕΙΣΑΓΩΓΗΣ (ΑΥΞΟΥΣΑ) Π[N]

```
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  J←I-1
  TEMP←Π[I]
  ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ
  ΟΣΟ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ J=0 ΤΟΤΕ
      ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ_ΑΝ TEMP<Π[J] ΤΟΤΕ
      Π[J+1]←Π[J]
      J←J-1
    ΑΛΛΙΩΣ
      ΒΡΕΘΗΚΕ← ΑΛΗΘΗΣ
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Π[J+1]←TEMP
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  J←I
  TEMP←Π[I]
  ΒΡΕΘ←ΑΛΗΘΗΣ
  ΟΣΟ J>1 ΚΑΙ ΒΡΕΘ=ΑΛΗΘΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ Π[J-1]>TEMP ΤΟΤΕ
      Π[J]←Π[J-1]
      J←J-1
    ΑΛΛΙΩΣ
      ΒΡΕΘ← ΨΕΥΔΗΣ
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Π[J]←TEMP
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ N
  J←I
  ΒΡΕΘ←ΑΛΗΘΗΣ
  ΟΣΟ J>1 ΚΑΙ ΒΡΕΘ=ΑΛΗΘΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ Π[J-1]>Π[J] ΤΟΤΕ
      TEMP←Π[J-1]
      Π[J-1]←Π[J]
      Π[J]←TEMP
      J←J-1
    ΑΛΛΙΩΣ
      ΒΡΕΘ← ΨΕΥΔΗΣ
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
I←1
ΟΣΟ I<=N ΕΠΑΝΑΛΑΒΕ
  ΑΝ I=1 ΤΟΤΕ
    I←I+1
  ΑΛΛΙΩΣ_ΑΝ Π[I]>=Π[I-1] ΤΟΤΕ
    TEMP←Π[I-1]
    Π[I-1]←Π[I]
    Π[I]←TEMP
    I←I-1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΚΕΝΟ← N DIV 2          !ΔΙΑΣΤΗΜΑ ΚΕΝΟ
ΟΣΟ ΚΕΝΟ>0 ΕΠΑΝΑΛΑΒΕ
  ΓΙΑ I ΑΠΟ ΚΕΝΟ ΜΕΧΡΙ N
    TEMP←Π[I]
    J←I
    ΒΡΕΘΗΚΕ←ΑΛΗΘΗΣ
    ΟΣΟ J> ΚΕΝΟ ΚΑΙ ΒΡΕΘΗΚΕ=ΑΛΗΘΗΣ ΕΠΑΝΑΛΑΒΕ
      ΑΝ Π[J-ΚΕΝΟ]>TEMP ΤΟΤΕ
        Π[J]← Π[J-ΚΕΝΟ]
        J←J - ΚΕΝΟ
      ΑΛΛΙΩΣ
        ΒΡΕΘΗΚΕ←ΨΕΥΔΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
Π[J]←TEMP  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
ΚΕΝΟ←ΚΕΝΟ DIV 2  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΑΞΙΝΟΜΗΣΗ ΓΡΑΜΜΗΣ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ Π[Ν,Μ]

```
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ Ν
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Μ
  ΓΙΑ J ΑΠΟ Μ ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
    ΑΝ Π[Κ, J-1] > Π[Κ, J] ΤΟΤΕ
      ΑΝΤΙΜΕΤΑΘΕΣΕ Π[J-1, Κ], Π[J, Κ]
      TEMP ← Π[Κ, J-1]
      Π[Κ, J-1] ← Π[Κ, J]
      Π[Κ, J] ← TEMP
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Για φθίνουσα ταξινόμηση έχουμε: Αν $\Pi[k, j-1] < \Pi[k, j]$ τότε

ΤΑΞΙΝΟΜΗΣΗ ΤΗΣ 1ΗΣ ΓΡΑΜΜΗΣ ΚΑΙ ΣΥΣΧΕΤΙΣΗ ΜΕ ΥΠΟΛΟΙΠΕΣ

```
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Μ
ΓΙΑ J ΑΠΟ Μ ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
  ΑΝ Π[1, J-1] > Π[1, J] ΤΟΤΕ
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ Ν
      TEMP ← Π[Κ, J-1]
      Π[Κ, J-1] ← Π[Κ, J]
      Π[Κ, J] ← TEMP
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΑΞΙΝΟΜΗΣΗ ΣΤΗΛΗΣ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ Π[Ν,Μ]

```
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ Μ
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν
ΓΙΑ J ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
  ΑΝ Π[J-1, Κ] > Π[J, Κ] ΤΟΤΕ
    ΑΝΤΙΜΕΤΑΘΕΣΕ Π[J-1, Κ], Π[J, Κ]
    TEMP ← Π[J, Κ-1]
    Π[J, Κ-1] ← Π[J, Κ]
    Π[J, Κ] ← TEMP
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Για φθίνουσα ταξινόμηση έχουμε: Αν $\Pi[j-1, κ] < \Pi[j, κ]$ τότε

ΤΑΞΙΝΟΜΗΣΗ ΤΗΣ 1ΗΣ ΣΤΗΛΗΣ ΚΑΙ ΣΥΣΧΕΤΙΣΗ ΜΕ ΥΠΟΛΟΙΠΕΣ

```
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν
ΓΙΑ J ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
  ΑΝ Π[J-1, 1] > Π[J, 1] ΤΟΤΕ
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ Μ
      TEMP ← Π[J-1, Κ]
      Π[J-1, Κ] ← Π[J, Κ]
      Π[J, Κ] ← TEMP
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΒΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΔΙΣΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ Π[N,M] ΟΠΟΥ Ν: ΓΡΑΜΜΕΣ, Μ: ΣΤΗΛΕΣ

ΕΝΕΡΓΕΙΕΣ ΣΕ ΟΛΟ ΤΟΝ ΠΙΝΑΚΑ

- **ΓΕΜΙΣΜΑ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΠΙΝΑΚΑ Ο[N] ΠΟΥ ΣΥΣΧΕΤΙΖΟΝΤΑΙ.**

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΔΙΑΒΑΣΕ Ο[Ι]
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
    ΔΙΑΒΑΣΕ Π[Ι, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- **ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ ΟΛΩΝ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N,M].**

```
Σ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
    Σ←Σ+Π[Ι, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Σ
```

- **ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ Π[N,M].**

```
ΠΛ←0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
    ΑΝ Π[Ι]=.... ΤΟΤΕ
      ΠΛ←ΠΛ+1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛ
```

- **ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ ΣΕ ΠΙΝΑΚΑ Π[N,M].**

```
ΜΑΧ←Π[1,1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
    ΑΝ Π[Ι, J]>ΜΑΧ ΤΟΤΕ
      ΜΑΧ←Π[Ι, J]
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ
```

- **ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ ΣΕ ΠΙΝΑΚΑ Π[N,M] ΜΕ ΚΑΘΟΡΙΣΜΟ ΘΕΣΗΣ.**

```
ΜΑΧ←Π[1,1]
ΓΡΑΜΜΗ←1
ΣΤΗΛΗ←1
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
    ΑΝ Π[Ι, J]>ΜΑΧ ΤΟΤΕ
      ΜΑΧ←Π[Ι, J]
      ΓΡΑΜΜΗ←Ι
      ΣΤΗΛΗ← J
```

Αν θέλουμε να γίνονται
οι ενέργειες με βάση
κάποια συγκεκριμένα
κριτήρια βάζουμε
ΑΝ <κριτήριο> ΤΟΤΕ

```
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΑΧ, ΓΡΑΜΜΗ, ΣΤΗΛΗ
```

• **ΕΥΡΕΣΗ ΜΕΓΙΣΤΩΝ ΣΕ ΠΙΝΑΚΑ Π[N,M] ΜΕ ΚΑΘΟΡΙΣΜΟ ΘΕΣΗΣ.**

```
ΜΑΧ←Π[1,1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
        ΑΝ Π[I, J]>ΜΑΧ ΤΟΤΕ
            ΜΑΧ←Π[I, J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
        ΑΝ Π[I, J]=ΜΑΧ ΤΟΤΕ
            ΓΡΑΨΕ ΜΑΧ, Ι, J
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• **ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΟΥ ΣΕ ΠΙΝΑΚΑ Π[N,M].**

```
ΜΙΝ←Π[1,1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
        ΑΝ Π[I, J]<ΜΙΝ ΤΟΤΕ
            ΜΙΝ←Π[I, J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΙΝ
```

• **ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΟΥ ΣΕ ΠΙΝΑΚΑ Π[N,M] ΜΕ ΚΑΘΟΡΙΣΜΟ ΘΕΣΗΣ.**

```
ΜΙΝ←Π[1,1]
ΓΡΑΜΜΗ←1
ΣΤΗΛΗ←1
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
        ΑΝ Π[I, J]<ΜΙΝ ΤΟΤΕ
            ΜΙΝ←Π[I, J]
            ΓΡΑΜΜΗ←I
            ΣΤΗΛΗ← J
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΙΝ, ΓΡΑΜΜΗ, ΣΤΗΛΗ
```

• **ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΩΝ ΣΕ ΠΙΝΑΚΑ Π[N,M] ΜΕ ΚΑΘΟΡΙΣΜΟ ΘΕΣΗΣ.**

```
ΜΙΝ←Π[1,1]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
        ΑΝ Π[I, J]<ΜΙΝ ΤΟΤΕ
            ΜΙΝ←Π[I, J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
```

```
    ΑΝ Π[Ι, J]=ΜΙΝ ΤΟΤΕ
      ΓΡΑΨΕ ΜΙΝ, Ι, J
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• **ΜΕΣΟΣ ΟΡΟΣ ΟΛΩΝ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΙΝΑΚΑ Π[Ν,Μ]**

```
SUM ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ
    SUM ← SUM+Π[Ι,J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ ← SUM/(Ν*Μ)
ΓΡΑΨΕ ΜΟ
```


ΕΝΕΡΓΕΙΕΣ ΑΝΑ ΓΡΑΜΜΗ ΤΟΥ ΠΙΝΑΚΑ

• ΓΕΜΙΣΜΑ ΠΙΝΑΚΑ Π[N,M] ΚΑΤΑ ΓΡΑΜΜΗ.

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
    ΔΙΑΒΑΣΕ Π[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΕΜΦΑΝΙΣΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΤΑ ΓΡΑΜΜΗ.

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
    ΓΡΑΨΕ Π[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ ΚΑΤΑ ΓΡΑΜΜΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N                                → ΓΡΑΜΜΗ
  Σ ← 0
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M                                → ΣΤΗΛΗ
    Σ ← Σ + Π[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΘ[I] ← Σ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ ΤΗΣ 5^{ΗΣ} ΓΡΑΜΜΗΣ ΠΙΝΑΚΑ Π[N,M].

```
Σ ← 0
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
  Σ ← Σ + Π[5, J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Σ
```

• ΕΥΡΕΣΗ ΜΕΓΙΣΤΩΝ ΚΑΤΑ ΓΡΑΜΜΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N                                → ΓΡΑΜΜΗ
  ΜΑΧ ← Π[I, 1]
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M                                → ΣΤΗΛΗ
    ΑΝ Π[I, J] > ΜΑΧ ΤΟΤΕ
      ΜΑΧ ← Π[I, J]
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΠΜΑΧ[I] ← ΜΑΧ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΩΝ ΚΑΤΑ ΓΡΑΜΜΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N                                → ΓΡΑΜΜΗ
  ΜΙΝ ← Π[I, 1]
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M                                → ΣΤΗΛΗ
    ΑΝ Π[I, J] < ΜΙΝ ΤΟΤΕ
      ΜΙΝ ← Π[I, J]
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΠΜΙΝ[I] ← ΜΙΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- **ΜΕΣΟΣ ΟΡΟΣ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΗΣ ΚΑΘΕ ΓΡΑΜΜΗΣ ΤΟΥ ΠΙΝΑΚΑ Π[N,M]**

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N                → ΓΡΑΜΜΗ
  SUM ← 0
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M                → ΣΤΗΛΗ
    SUM ← SUM + Π[I,J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΜΟ ← SUM/M
  ΓΡΑΨΕ ΜΟ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- **ΥΠΟΛΟΓΙΣΜΟΣ ΜΟ ΑΝΑ ΓΡΑΜΜΗ ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΤΟΥ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.**

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ M
  SUM ← 0
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ N
    SUM ← SUM + Π [I,J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΜΟ [I] ← SUM / N
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΕΝΕΡΓΕΙΕΣ ΑΝΑ ΣΤΗΛΗ ΤΟΥ ΠΙΝΑΚΑ

• ΓΕΜΙΣΜΑ ΠΙΝΑΚΑ Π[N,M] ΚΑΤΑ ΣΤΗΛΗ.

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
    ΔΙΑΒΑΣΕ Π[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΕΜΦΑΝΙΣΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΤΑ ΣΤΗΛΗ.

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M
  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
    ΓΡΑΨΕ Π[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ ΚΑΤΑ ΣΤΗΛΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M           → ΣΤΗΛΗ
  Σ←0
  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N         → ΓΡΑΜΜΗ
    Σ←Σ+Π[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΘ[K]←Σ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ ΤΗΣ 3^{ΗΣ} ΣΤΗΛΗΣ ΠΙΝΑΚΑ Π[N,M].

```
Σ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
  Σ←Σ+Π[I,3]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Σ
```

• ΕΥΡΕΣΗ ΜΕΓΙΣΤΩΝ ΚΑΤΑ ΣΤΗΛΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M           → ΣΤΗΛΗ
  ΜΑΧ←Π[1, J]
  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N         → ΓΡΑΜΜΗ
    ΑΝ Π[I, J]>ΜΑΧ ΤΟΤΕ
      ΜΑΧ←Π[I, J]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΜΑΧ[J]←ΜΕΓ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

• ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΩΝ ΚΑΤΑ ΣΤΗΛΗ ΠΙΝΑΚΑ Π[N,M] ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M           → ΣΤΗΛΗ
  ΜΙΝ←Π[1, J]
  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N         → ΓΡΑΜΜΗ
    ΑΝ Π[I, J]<ΜΙΝ ΤΟΤΕ
      ΜΙΝ←Π[I, J]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
    ΠΜΙΝ[J]←ΜΙΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- **ΜΕΣΟΣ ΟΡΟΣ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΗΣ ΚΑΘΕ ΣΤΗΛΗΣ ΤΟΥ ΠΙΝΑΚΑ Π[N,M]**

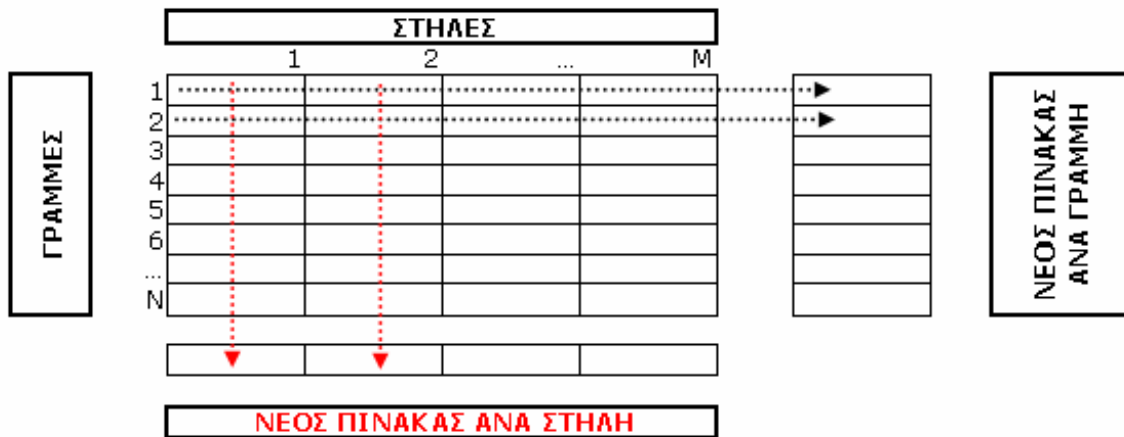
```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ           → ΣΤΗΛΗ
    SUM ← 0
    ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν       → ΓΡΑΜΜΗ
        SUM ← SUM+Π[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ ← SUM/Ν
    ΓΡΑΨΕ ΜΟ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- **ΥΠΟΛΟΓΙΣΜΟΣ ΜΟ ΑΝΑ ΣΤΗΛΗ ΚΑΙ ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΤΟΥ ΣΕ ΞΕΧΩΡΙΣΤΟ ΠΙΝΑΚΑ.**

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Ν
    SUM ← 0
    ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Μ
        SUM ← SUM + Π[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ [J] ← SUM / Ν
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΣΥΝΟΠΤΙΚΑ!

ΑΡΧΙΚΟΠΟΙΗΣΗ ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ ΓΡΑΜΜΗ ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΣΤΗΛΗ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΑΞΙΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΠΙΝΑΚΑ	ΕΠΕΞΕΡΓΑΣΙΑ ΟΛΟΚΛΗΡΟΥ ΤΟΥ ΠΙΝΑΚΑ ΚΑΤΑ ΓΡΑΜΜΕΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ ΓΡΑΜΜΗ ΑΡΧΙΚΟΠΟΙΗΣΗ ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΣΤΗΛΗ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΑΞΙΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΓΡΑΜΜΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΕΠΕΞΕΡΓΑΣΙΑ ΓΙΑ ΚΑΘΕ ΓΡΑΜΜΗ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΣΤΗΛΗ ΑΡΧΙΚΟΠΟΙΗΣΗ ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ ΓΡΑΜΜΗ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΑΞΙΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΤΗΛΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΕΠΕΞΕΡΓΑΣΙΑ ΓΙΑ ΚΑΘΕ ΣΤΗΛΗ



ΤΕΤΡΑΓΩΝΙΚΟΣ ΠΙΝΑΚΑΣ

Ένας πίνακας ονομάζεται τετραγωνικός, όταν έχει ίσο αριθμό γραμμών και στηλών. Σε ένα τέτοιο πίνακα ορίζουμε δύο διαγώνιους, την κύρια και την δευτερεύουσα

Κύρια διαγώνιος

X			
	X		
		X	
			X

Συνθήκη στοιχείων κύριας διαγωνίου:

$$\text{γραμμή} = \text{στήλη}$$

Συνθήκη στοιχείων πάνω από την κύρια διαγώνιο:

$$\text{γραμμή} < \text{στήλη}$$

Συνθήκη στοιχείων κάτω από την κύρια διαγώνιο:

$$\text{γραμμή} > \text{στήλη}$$

Δευτερεύουσα διαγώνιος

			X
		X	
	X		
X			

Συνθήκη στοιχείων δευτερεύουσας διαγωνίου:

$$\text{γραμμή} + \text{στήλη} = N + 1$$

Συνθήκη στοιχείων πάνω από την δευτερεύουσα διαγώνιο:

$$\text{γραμμή} + \text{στήλη} < N + 1$$

Συνθήκη στοιχείων κάτω από την δευτερεύουσα διαγώνιο:

$$\text{γραμμή} + \text{στήλη} > N + 1$$

Όταν ένας τετραγωνικός πίνακας έχει περιττό αριθμό από γραμμές και στήλες, τότε το κεντρικό στοιχείο ανήκει και στις δύο διαγώνιους.

Τριγωνικός Άνω: Όλα τα στοιχεία κάτω από την κύρια διαγώνιο είναι μηδέν.

Τριγωνικός Κάτω: Όλα τα στοιχεία πάνω από την κύρια διαγώνιο είναι μηδέν.

ΣΤΟΙΒΑ

- **ΏΘΗΣΗ ΕΝΟΣ ΔΕΔΟΜΕΝΟΥ**

```
ΔΙΑΒΑΣΕ Χ
ΑΝ ΤΟΡ<Ν ΤΟΤΕ
    ΤΟΡ←ΤΟΡ + 1
    TABLE[ΤΟΡ] ←Χ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΣΤΟΙΒΑ ΓΕΜΑΤΗ'
ΤΕΛΟΣ_ΑΝ
```

- **ΏΘΗΣΗ ΠΟΛΛΩΝ ΔΕΔΟΜΕΝΩΝ**

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Χ
ΑΝ ΤΟΡ<Ν ΤΟΤΕ
    ΤΟΡ←ΤΟΡ + 1
    TABLE[ΤΟΡ] ←Χ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΣΤΟΙΒΑ ΓΕΜΑΤΗ'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ ΤΟΡ=Ν
```

- **ΑΠΩΘΗΣΗ ΕΝΟΣ ΔΕΔΟΜΕΝΟΥ**

```
ΑΝ ΤΟΡ>=1 ΤΟΤΕ
    Χ←TABLE[ΤΟΡ]
    ΤΟΡ←ΤΟΡ - 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΣΤΟΙΒΑ ΑΔΕΙΑ'
ΤΕΛΟΣ_ΑΝ
```

- **ΑΠΩΘΗΣΗ ΠΟΛΛΩΝ ΔΕΔΟΜΕΝΩΝ**

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΤΟΡ>=1 ΤΟΤΕ
    Χ←TABLE[ΤΟΡ]
    ΤΟΡ←ΤΟΡ - 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΣΤΟΙΒΑ ΑΔΕΙΑ'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ ΤΟΡ=0
```

Κενή στοίβα: ΤΟΡ←0
Απώθηση: ΤΟΡ←ΤΟΡ - 1
Ώθηση: ΤΟΡ←ΤΟΡ + 1
Γεμάτη στοίβα: ΤΟΡ←Ν

ΟΥΡΑ

- **ΕΙΣΑΓΩΓΗ ΕΝΟΣ ΔΕΔΟΜΕΝΟΥ**

```
ΔΙΑΒΑΣΕ X
AN REAR<N TOTE
    REAR←REAR + 1
    TABLE[REAR] ←X
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΟΥΡΑ ΓΕΜΑΤΗ ΣΤΟ ΠΙΣΩ ΜΕΡΟΣ'
ΤΕΛΟΣ_ΑΝ
```

- **ΕΙΣΑΓΩΓΗ ΠΟΛΛΩΝ ΔΕΔΟΜΕΝΩΝ**

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ X
AN REAR<N TOTE
    REAR←REAR + 1
    TABLE[REAR] ←X
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΟΥΡΑ ΓΕΜΑΤΗ ΣΤΟ ΠΙΣΩ ΜΕΡΟΣ'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ REAR=N
```

- **ΕΞΑΓΩΓΗ ΕΝΟΣ ΔΕΔΟΜΕΝΟΥ**

```
AN FRONT <= REAR TOTE
    X←TABLE[FRONT]
    FRONT←FRONT + 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΟΥΡΑ ΑΔΕΙΑ'
ΤΕΛΟΣ_ΑΝ
```

- **ΕΞΑΓΩΓΗ ΠΟΛΛΩΝ ΔΕΔΟΜΕΝΩΝ**

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
AN FRONT <= REAR TOTE
    X←TABLE[FRONT]
    FRONT←FRONT + 1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΟΥΡΑ ΑΔΕΙΑ'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ FRONT > REAR
```

Κενή ουρά: front←0, rear←0
Γεμάτη ουρά: front←1, rear←N
Ένα στοιχείο: front=rear (<>0)
Γεμάτη πίσω μέρος: front>1, rear←N
Εισαγωγή: rear←rear+1
Εξαγωγή: front←front+1
Πλήθος στοιχείων ουράς: rear-front+1