Κεφάλαια 3, 9 Δομές Δεδομένων - Πίνακες

(§3.1) Δεδομένα

Η Πληροφορική μελετά τα δεδομένα από τις σκοπιές:

- Υλικού: αποθήκευση στη μνήμη με διάφορες αναπαραστάσεις π.χ. ASCII
- Γλωσσών προγραμματισμού: π.χ. οι διαφορετικοί τύποι των μεταβλητών
- Δομών Δεδομένων: π.χ. αρχεία με εγγραφές και πεδία
- Ανάλυσης Δεδομένων: καταγραφή και αλληλοσυσχέτιση των δεδομένων ώστε να αναπαρασταθεί η γνώση. π.χ. Βάσεις Δεδομένων

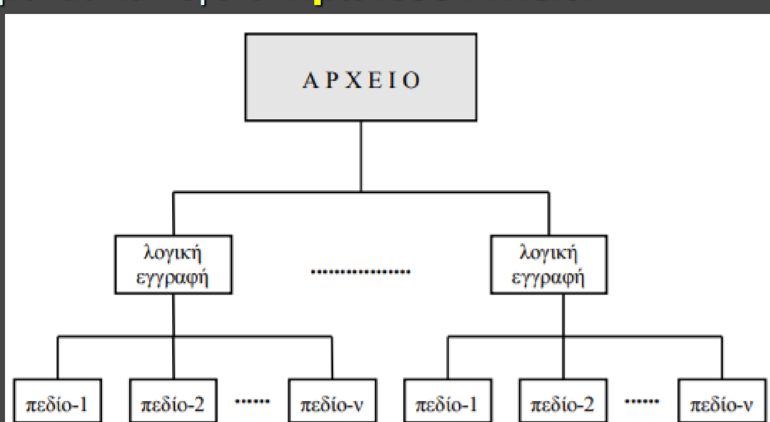
(§3.2) Δομή δεδομένων (Δ.Δ.)

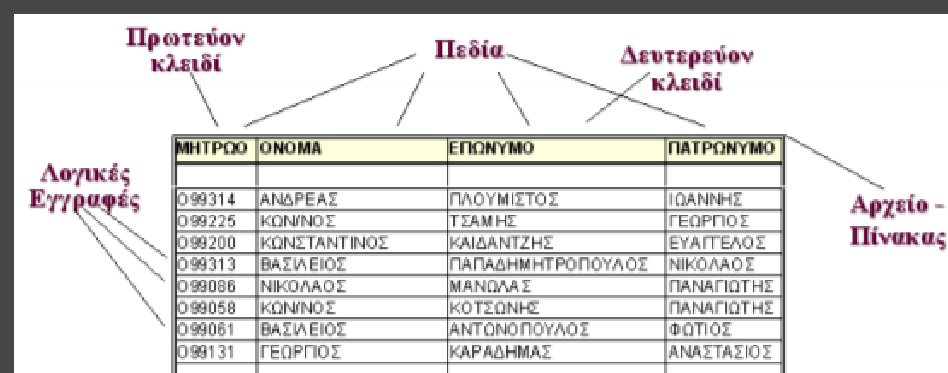
Δομές Δεδομένων δευτερεύουσας μνήμης

ΟΡΙΣΜΟΣ

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

Σε μεγάλες 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) όπου τα δεδομένα δεν χάνονται, αν κλείσει η εφαρμογή ή ο ΗΥ σε αντίθεση με τα δεδομένα της κύριας μνήμης. Τα στοιχεία ενός αρχείου ονομάζονται **εγγραφές** (records), όπου κάθε εγγραφή αποτελείται από ένα ή περισσότερα **πεδία** (fields). Το πεδίο μίας εγγραφής που την ταυτοποιεί μοναδικά λέγεται **πρωτεύον κλειδί**.





ΜΗΤΡΩΟ	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΠΑΤΡΩΝΥΜΟ
099314	ΑΝΔΡΕΑΣ	ΠΛΟΥΜΙΣΤΟΣ	ΙΩΑΝΝΗΣ
099225	ΚΩΝ/ΝΟΣ	ΤΣΑΜΗΣ	ΓΕΩΡΓΙΟΣ
099200	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΚΑΙΔΑΝΤΖΗΣ	ΕΥΑΓΓΕΛΟΣ
099313	ΒΑΣΙΛΕΙΟΣ	ΠΑΠΑΔΗΜΗΤΡΟΠΟΥΛΟΣ	ΝΙΚΟΛΑΟΣ
099086	ΝΙΚΟΛΑΟΣ	ΜΑΝΩΛΑΣ	ΠΑΝΑΓΙΩΤΗΣ
099058	ΚΩΝ/ΝΟΣ	ΚΟΤΣΩΝΗΣ	ΠΑΝΑΓΙΩΤΗΣ
099061	ΒΑΣΙΛΕΙΟΣ	ΑΝΤΩΝΟΠΟΥΛΟΣ	ΦΩΤΙΟΣ
099131	ΓΕΩΡΓΙΟΣ	ΚΑΡΑΔΗΜΑΣ	ΑΝΑΣΤΑΣΙΟΣ

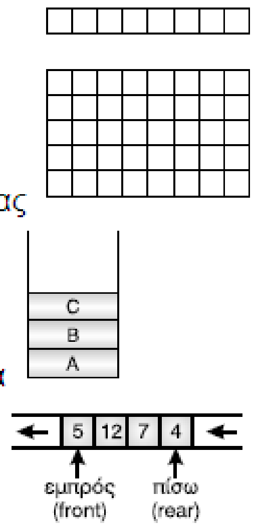
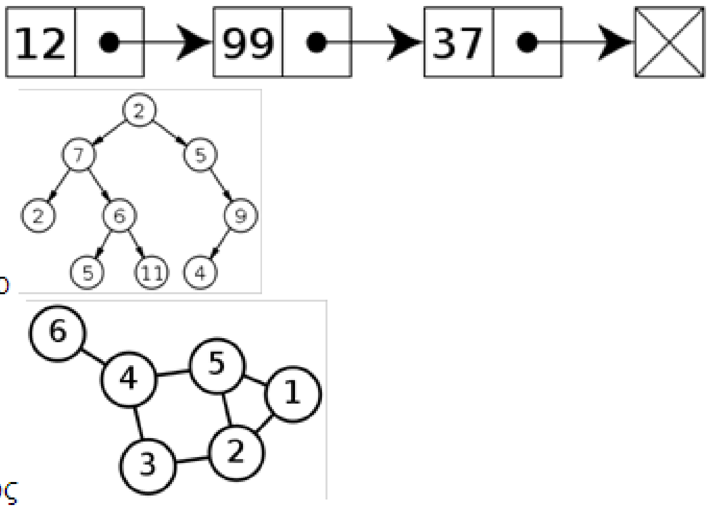
(§3.2) Βασικές λειτουργίες επί των δομών δεδομένων

1. **Προσπέλαση** (access), πρόσβαση σε ένα κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχόμενό του.
2. **Εισαγωγή** (insertion), δηλαδή η προσθήκη νέων κόμβων σε μία υπάρχουσα δομή.
3. **Διαγραφή** (deletion), που αποτελεί το αντίστροφο της εισαγωγής, δηλαδή ένας κόμβος αφαιρείται από μία δομή.
4. **Αναζήτηση** (searching), κατά την οποία προσπελαύνονται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι που έχουν μια δεδομένη ιδιότητα.
5. **Ταξινόμηση** (sorting), όπου οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά.
6. **Αντιγραφή** (copying), κατά την οποία όλοι οι κόμβοι ή μερικοί από τους κόμβους μίας δομής αντιγράφονται σε μία άλλη δομή.
7. **Συγχώνευση** (merging), κατά την οποία δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
8. **Διαχωρισμός** (separation), που αποτελεί την αντίστροφη πράξη της συγχώνευσης.

Η Εισαγωγή και η Διαγραφή <u>δεν</u> εφαρμόζονται στις στατικές Δ.Δ.

Εξίσωση του Wirth : Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα

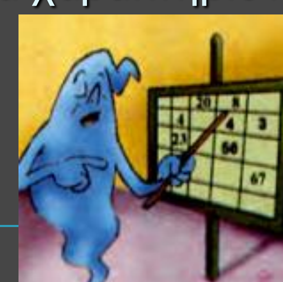
Κατηγορίες δομών δεδομένων

Στατικές	Δυναμικές
ΔΔ σταθερού και προκαθορισμένου μεγέθους (κατά τη μεταγλώττιση). Αποθηκεύονται σε **συνεχόμενες** θέσεις μνήμης.	ΔΔ δυναμικά μεταβαλλόμενου μεγέθους (κατά την εκτέλεση). Αποθηκεύονται σε **μη συνεχόμενες** θέσεις μνήμης με τη μέθοδο της δυναμικής παραχώρησης μνήμης
+ απλότητα προγραμματιστικής χρήσης	- πολυπλοκότητα προγραμματιστικής χρήσης
- μη ευέλικτα προγράμματα	+ ευελιξία προγραμμάτων
- σπατάλη μνήμης	+ εξοικονόμηση μνήμης
Παραδείγματα:  • Πίνακας • Στοίβα • Ουρά	Παραδείγματα:  • Λίστα • Δέντρο • Γράφος

(§3.3 - §9.1) Ορισμός: πίνακας

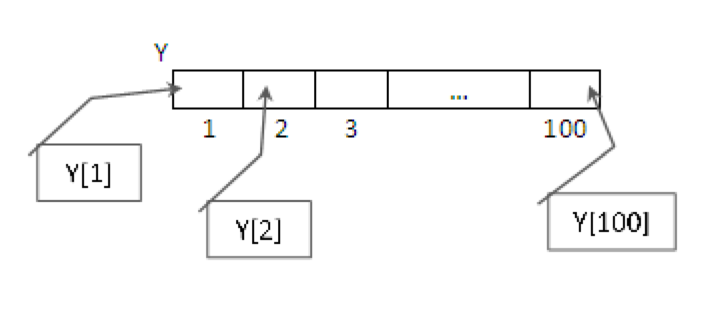
Στατική ΔΔ κατάλληλη για την προσωρινή αποθήκευση ενός συνόλου τιμών τιμών γνωστού πλήθους (ή γνωστού μέγιστου πλήθους), του ίδιου τύπου. Βασικά χαρακτηριστικά:

- Όνομα
- Τύπος (Ακέραιος / Πραγματικός / Χαρακτήρες / Λογικός)
- Διαστάσεις: 1, 2, 3, ...
- Μέγεθος ανά διάσταση



(§9.3) Δήλωση – παραδείγματα

α) Μονοδιάστατος (1-Δ)
π.χ. ύψη 100 μαθητών
Μεταβλητές
Πραγματικές: Υ[100] ⇒

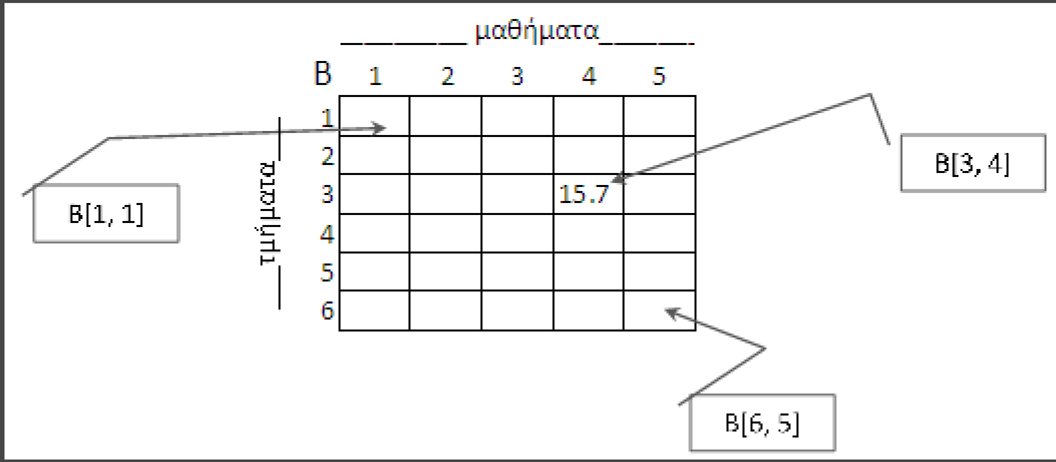


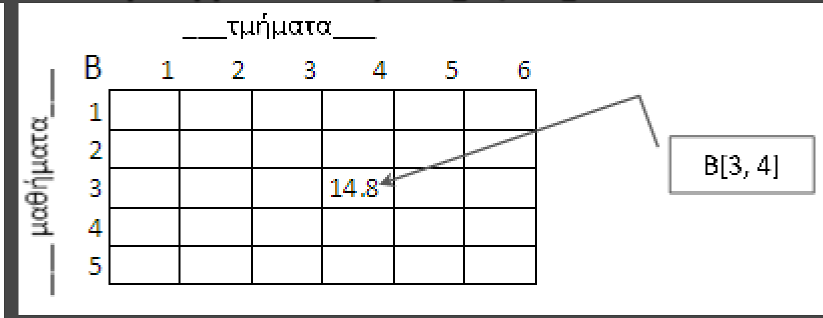
β) Δισδιάστατος (2-Δ)
π.χ. ΜΟ βαθμών 6 τμημάτων της Γ' Λυκείου σε 5 μαθήματα

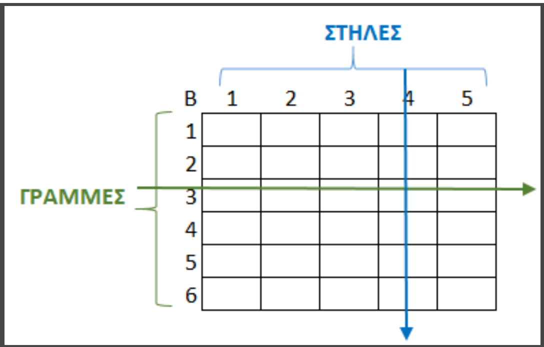
Μεταβλητές
Πραγματικές: Β[6, 5]

ή

Μεταβλητές
Πραγματικές: Β[5, 6]

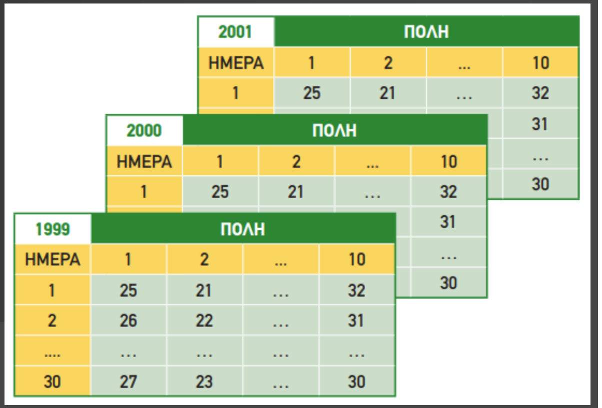




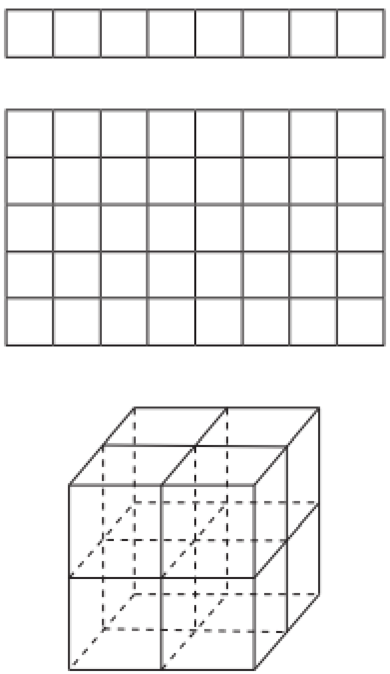


Πλήθος κελιών = 6 x 5 = 30

(§9.3) Δήλωση – παραδείγματα
γ) Τρισδιάστατος (3-Δ)
π.χ. ΜΟ βαθμών 6 τμημάτων της Γ’ Λυκείου σε 5 μαθήματα για 2 τετράμηνα



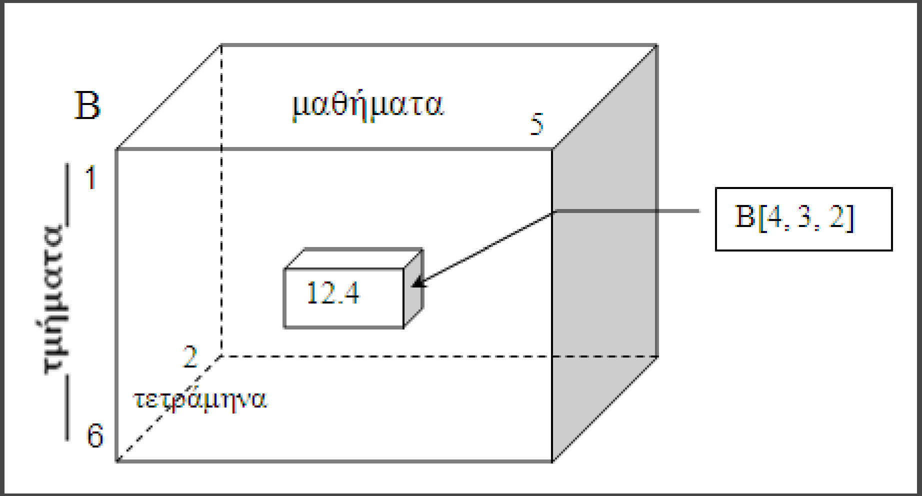
1999	ΠΟΛΗ			
ΗΜΕΡΑ	1	2	...	10
1	25	21	…	32
2	26	22	…	31
....	…	…	…	…
30	27	23	…	30



Σχ. 3.1. Παραδείγματα πινάκων (μονοδιάστατος, δισδιάστατος, τρισδιάστατος)

Μεταβλητές

Πραγματικές: B[6, 5, 2]



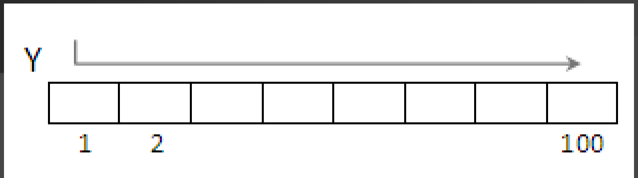
Πλήθος κελιών = 6 x 5 x 2 = 60

Παραδείγματα 2Δ πινάκων

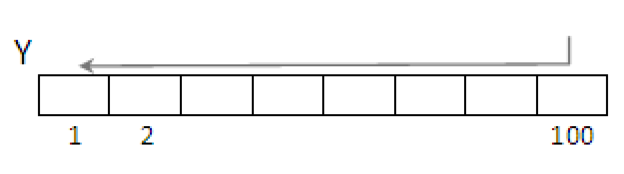
1. εισπράξεις 100 επιχειρήσεων για 24 ώρες (ΕΙΣ[100, 24]), 7 ημέρες μιας εβδομάδας (ΕΙΣ[100, 7]), 360 ημέρες ενός έτους με κάθε μήνα από 30 ημέρες (ΕΙΣ[100, 360]), 12 μήνες ενός έτους (ΕΙΣ[100, 12]), 10 έτη (ΕΙΣ[100, 10])
2. εκλογές με 200 μαθητές να ψηφίζουν το πολύ 15 από 50 υποψήφιους (Ψ[200, 50])
3. βαθμοί 100 μαθητών σε 10 μαθήματα (Β[100, 10])
4. ρίψεις/άλματα 100 αθλητών σε 6 προσπάθειες (ΕΠ[100, 6])
5. χρόνοι γύρου 10 πιλότων F1 σε αγώνα 50 γύρων (Χ[10, 50])
6. απαντήσεις 100 διαγωνιζόμενων σε 30 ερωτήσεις Σ/Λ ή πολλαπλής επιλογής (Α[100, 30])
7. κρατήσεις θέσεων σε θέατρο 20 σειρών των 30 καθισμάτων (Θ[20, 30])
8. αποτελέσματα (Ν/Ι/Η) αγώνων 16 ομάδων με μονούς/διπλούς αγώνες (Α[16, 16])
9. χιλιομετρικές αποστάσεις 30 πόλεων (Χ[30, 30])
10. βαθμολογίες σε διαγωνισμό με συμμετοχή 20 χωρών (Β[20, 20])
11. διασυνδέσεις (0/1) 1000 μελών κοινωνικού δικτύου (Φ[1000, 1000])
12. συμβατότητα (0/1) μεταξύ 8 ομάδων αίματος (Σ[8,8])
13. ανταποκρίσεις (0/1) μεταξύ 30 αεροδρομίων (Α[30, 30])
14. ατομικά στοιχεία 200 υπαλλήλων: 1. όνομα, 2. επώνυμο, 3. email, 4. διεύθυνση κατοικίας (Σ[200, 4])
15. εικονοστοιχεία (pixel 0/1) εικόνας 500x300 (Ι[500, 300])
16. αριθμός απασχολούμενων σε κτίριο με 10 ορόφους και 20 γραφεία ανά όροφο (ΕΡ[10, 20])
17. μεταφραστικό λεξικό δύο γλωσσών και για 30000 όρους (Λ[30000, 2])
18. χαρτογράφηση έκτασης π.χ. φάση ποδοσφαιρικού αγώνα σε γήπεδο 100x70 (Γ[100, 70]) με τοποθέτηση παικτών
19. λίστα 1000 τηλεφωνικών επαφών: 1. όνομα, 2. τηλέφωνο (ΕΠ[1000, 2])
20. παιχνίδια: σκάκι, ντάμα, sudoku, ναυμαχία, ναρκαλιευτής, τρίλιζα
21. τυχερά παιχνίδια: 100 τελευταίες κληρώσεις 6 τυχερών αριθμών του ΛΟΤΤΟ (Τ[100, 6])
22. ημερομηνίες: εορτολόγιο/ιστορικά γεγονότα/ημερομηνίες γέννησης (100) σε τριάδες: 1. ημέρα, 2. μήνας, 3. έτος (ΗΜ[100,3])
23. χρόνοι τερματισμού (ώρες, λεπτά, δευτερόλεπτα) 100 μαραθωνοδρόμων (Χ[100, 3])
24. 100 ρίψεις 2 ζαριών (Ζ[100, 2])
25. ωρολόγιο εβδομαδιαίο σχολικό πρόγραμμα (Π[7, 5])
26. διαθεσιμότητα για συνάντηση (0/1) 10 συνεργατών για κάθε ώρα ενός 24ώρου (Δ[10, 24])

Σάρωση κελιών πίνακα

α) Μονοδιάστατος (1-Δ) - π.χ. ύψη 100 μαθητών



για i από 1 μέχρι 100
  Αναφορά (Διάβασε/Γράψε/ ←) στο κελί Y[i]
ΤέλοςΕπανάληψης



για i από 100 μέχρι 1 μεβήμα -1
  Αναφορά (Διάβασε/Γράψε/ ←) στο κελί Y[i]
ΤέλοςΕπανάληψης

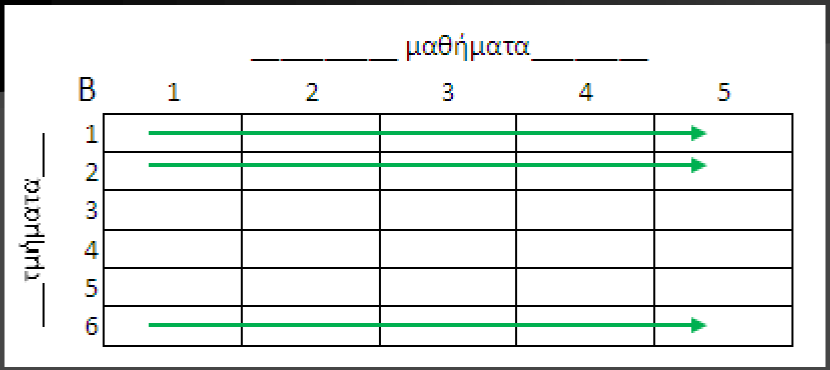
ή

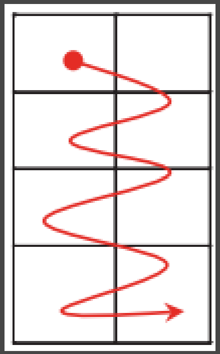
για i από 1 μέχρι 100
  Αναφορά (Διάβασε/Γράψε/ ←) στο κελί Y[101 - i]
ΤέλοςΕπανάληψης

Σάρωση κελιών πίνακα

β) Δισδιάστατος (2-Δ) - π.χ. ΜΟ βαθμών 6 τμημάτων της Γ' Λυκείου σε 5 μαθήματα

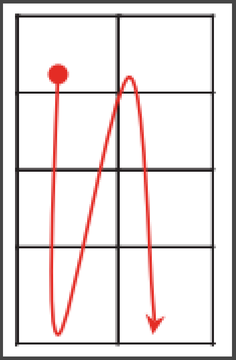
i. Κατά γραμμές:





για i από 1 μέχρι 6
  για j από 1 μέχρι 5
    Αναφορά (Διάβασε/Γράψε/ ←) στο κελί B[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης

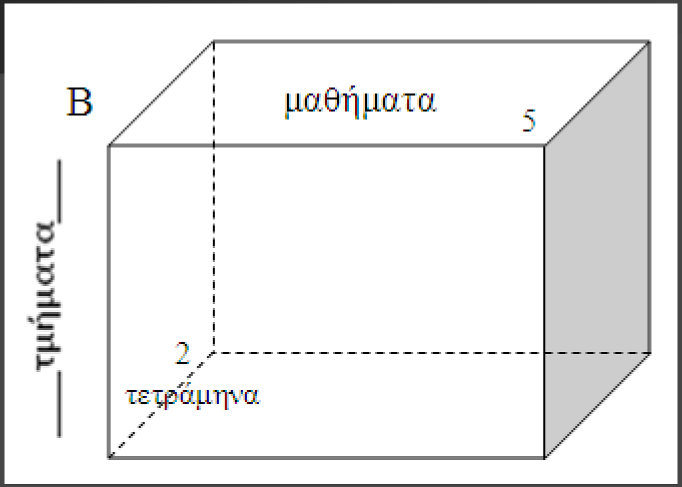
ii. Κατά στήλες:



για j από 1 μέχρι 5
  για i από 1 μέχρι 6
    Αναφορά (Διάβασε/Γράψε/ ←) στο κελί B[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης

Σάρωση κελιών πίνακα

γ) Τρισδιάστατος (3-Δ) - π.χ. ΜΟ βαθμών 6 τμημάτων της Γ΄ Λυκείου σε 5 μαθήματα για 2 τετράμηνα




```
για i από 1 μέχρι 6
  για j από 1 μέχρι 5
    για k από 1 μέχρι 2
      Αναφορά (Διάβασε/Γράψε/ ←) στο κελί B[i, j, k]
    ΤέλοςΕπανάληψης
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Αναφορές σε κελιά πίνακα

A	5	-1	7	1	3	12	-6	4
	1	2	3	4	5	6	7	8

Γράψε A[4] → 1

x ← 5 Γράψε A[x] → 3

Γράψε A[x+2] → -6

Γράψε A[x]+2 → 5

Γράψε A[A[x]] → 7

Γράψε A[A[x]+5] → 4

Γράψε A[A[2]+A[3]] → 12

Γράψε A[A[A[5]]] → -6

Γράψε A[A[2]+A[4]] → A[0]?

Γράψε A[A[6]-A[5]] → A[9]?

Αναφορές σε κελιά πίνακα

B	1	2	3	4	5
1	0	7	6	5	4
2	8	1	14	3	13
3	9	10	2	11	12

Γράψε B[1, 3] ➔ 6

x ← 3 Γράψε B[x, x] ➔ 2

Γράψε B[x, x+2] ➔ 12

Γράψε B[x, 1]+2 ➔ 11

Γράψε B[B[x, x], 5] ➔ 13

Γράψε B[B[x, x]+1, 5] ➔ 12

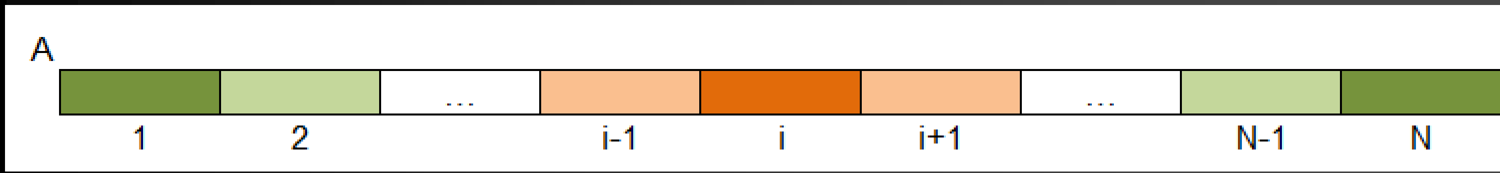
Γράψε B[B[2,2]+B[3,3], 2] ➔ 10

Γράψε B[B[B[2,4], 3], 4] ➔ 3

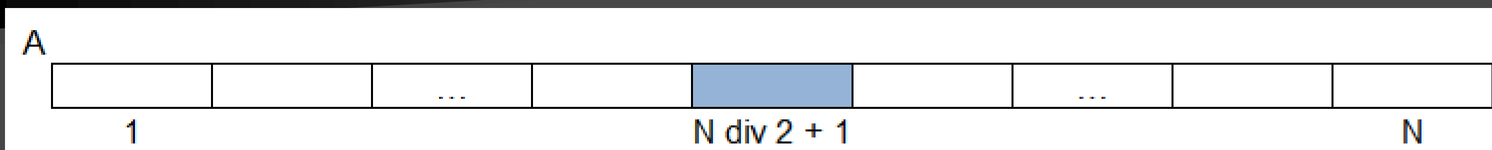
Γράψε B[B[2,2]-B[3,3], 1] ➔ B[-1,1]?

Γράψε B[1, B[3,3]+B[1,5]] ➔ B[1, 6]?

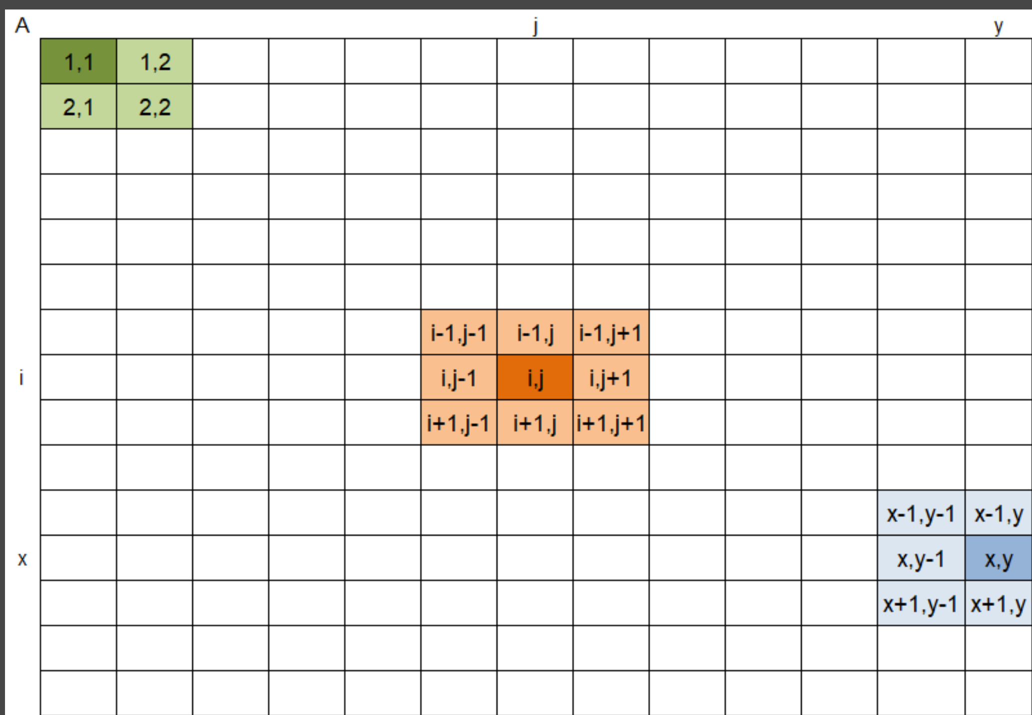
Γειτονικά κελιά 1-Δ πίνακα: 1/2



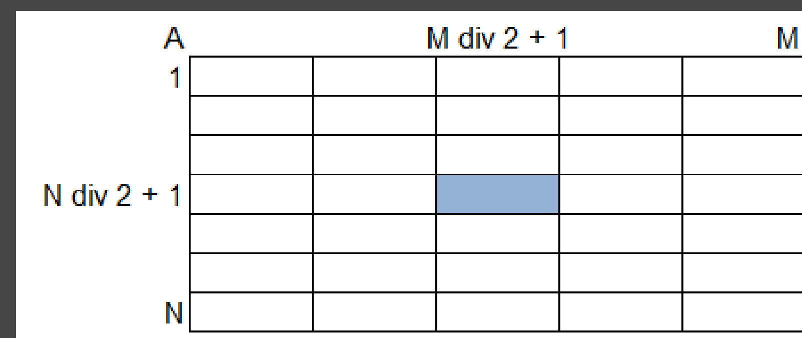
Μεσαίο κελί 1-Δ πίνακα A[N], N περιττός



Γειτονικά κελιά 2-Δ πίνακα: 3/5/8



Μεσαίο κελί 2-Δ πίνακα A[N, M], N, M περιττοί



Έστω 2 στοιχεία A[γ1,σ1] και A[γ2,σ2] πίνακα A[N,M]

- στοιχεία ίδιας γραμμής: γ1=γ2
- στοιχεία ίδιας στήλης: σ1=σ2
- στοιχεία ίδιας διαγωνίου: |γ1-γ2|=|σ1-σ2|

Γειτονικά κελιά 2-Δ πίνακα: 3/5/8

i-1,j-1	i-1,j	i-1,j+1
i,j-1	i,j	i,j+1
i+1,j-1	i+1,j	i+1,j+1

Εισαγωγή των συντεταγμένων (i, j) ενός κελιού πίνακα A[50, 100]
και εμφάνιση της τιμής του και των γειτονικών του κελιών

```
ΑρχήΕπανάληψης
  Διάβασε i, j
ΜέχριςΌτου i >= 1 ΚΑΙ i <= 50 ΚΑΙ j >= 1 ΚΑΙ j <= 100 ! εντός ορίων
για x από i-1 μέχρι i+1 ! σάρωση του 3x3 υπο-πίνακα με κέντρο το (i, j)
  για y από j-1 μέχρι j+1
    Αν x >= 1 ΚΑΙ x <= 50 ΚΑΙ y >= 1 ΚΑΙ y <= 100 τότε ! εντός ορίων
      Γράψε A[x, y]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

(§9.4) Τυπικές επεξεργασίες πινάκων:

1. Αθροίσματα *(ακέραιοι, πραγματικοί)*
2. Μέγιστα – ελάχιστα *(ακέραιοι, πραγματικοί, χαρακτήρες)*
3. Ταξινόμηση *(ακέραιοι, πραγματικοί, χαρακτήρες)*
4. Αναζήτηση *(ακέραιοι, πραγματικοί, χαρακτήρες, λογικοί)*
5. Συγχώνευση *(ακέραιοι, πραγματικοί, χαρακτήρες)*

Περιπτώσεις παραβίασης της καθοριστικότητας :

1. Διαίρεση με το μηδέν
2. Αρνητικό υπόριζο
3. Αναφορά έκφρασης σε απροσδιόριστη μεταβλητή/στοιχείο πίνακα
4. Παραβίαση ορίων πίνακα (1-Δ, 2-Δ, …)

Π.χ. 1 (για Α[100]):

Διάβασε x
Γράψε Α[x] ! $x \in [1, 100]$?

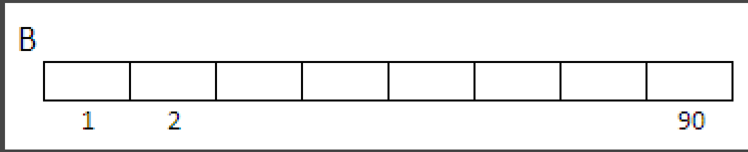
Π.χ. 2 (για Β[50, 100]):

Διάβασε x, y
Γράψε Β[x, y] ! $x \in [1, 50]$? ΚΑΙ $y \in [1, 100]$?

Επεξεργασία 1-Δ πίνακα

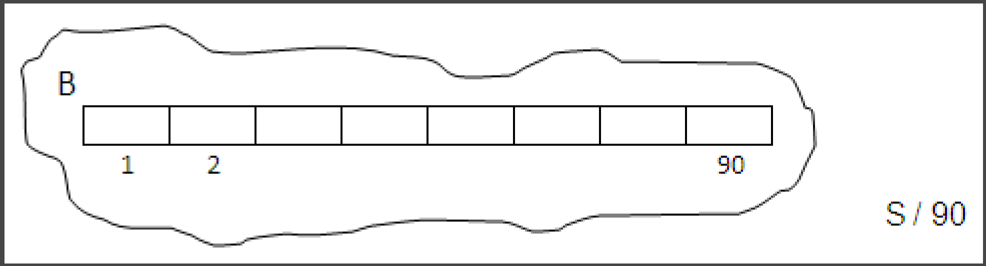
Πρόγραμμα το οποίο:
Διαβάζει σε κατάλληλο πίνακα τους βαθμούς 90 μαθητών σε ένα διαγώνισμα στην 20θμια κλίμακα (0-20), με έλεγχο εγκυρότητας.

```
για i από 1 μέχρι 90
  ΑρχήΕπανάληψης
    Διάβασε B[i]
  ΜέχριςΌτου B[i] >= 0  ΚΑΙ B[i] <= 20
ΤέλοςΕπανάληψης
```



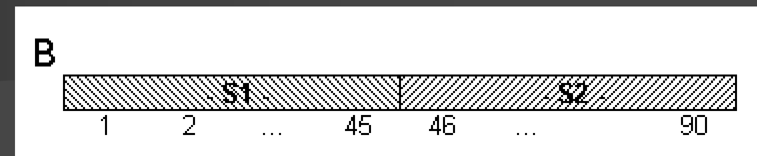
Βρίσκει τον μέσο όρο όλων των μαθητών.

```
S ← 0
για i από 1 μέχρι 90
  S ← S + B[i]
ΤέλοςΕπανάληψης
MO ← S / 90
Γράψε MO
```

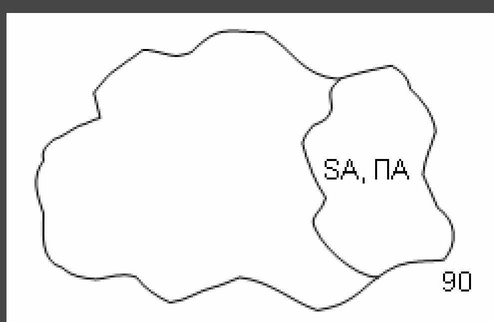


Βρίσκει τον μέσο όρο των 45 πρώτων και των 45 τελευταίων μαθητών.

```
S1 ← 0
S2 ← 0
για i από 1 μέχρι 45
  S1 ← S1 + B[i]
  S2 ← S2 + B[45 + i]
ΤέλοςΕπανάληψης
ΜΟ1 ← S1 / 45
ΜΟ2 ← S2 / 45
Γράψε ΜΟ1, ΜΟ2
```

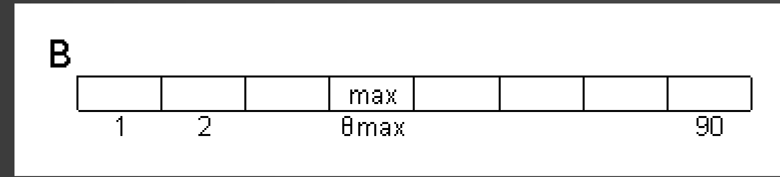


Βρίσκει τον μέσο όρο των άριστων (>18) μαθητών.




```
SA ← 0
ΠΑ ← 0
για i από 1 μέχρι 90
  Αν B[i] > 18 τότε
    SA ← SA + B[i]
    ΠΑ ← ΠΑ + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν ΠΑ <> 0 τότε
  ΜΟΑ ← SA / ΠΑ
  Γράψε ΜΟΑ
Αλλιώς
  Γράψε ‘κανένας’
ΤέλοςΑν
```

Βρίσκει τον μεγαλύτερο βαθμό και τον αριθμό (1-90) του μαθητή που τον έχει (χωρίς ισοτιμία).

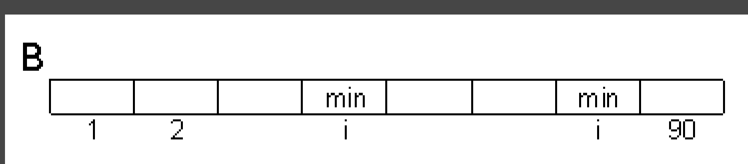



```
max ← B[1]
θmax ← 1
για i από 2 μέχρι 90
  Αν B[i] > max τότε
    max ← B[i]
    θmax ← i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε max, θmax
```

ή

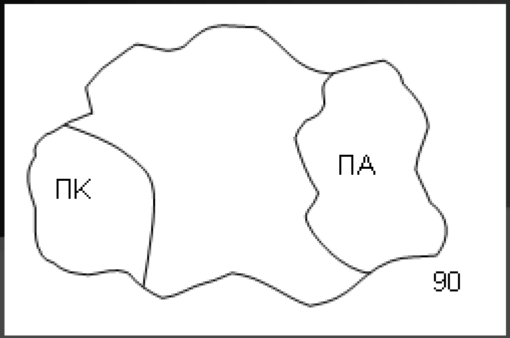
```
θmax ← 1
για i από 2 μέχρι 90
  Αν B[i] > B[θmax] τότε
    θmax ← i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε B[θmax], θmax
```

Βρίσκει τον μικρότερο βαθμό και τον αριθμό του μαθητή(ών) που τον έχει (με ισοτιμία)



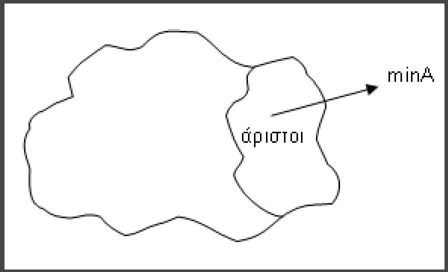

```
min ← B[1]
για i από 2 μέχρι 90
  Αν B[i] < min τότε
    min ← B[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε min
για i από 1 μέχρι 90
  Αν B[i] = min τότε
    Γράψε i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Βρίσκει τα % ποσοστά των «κακών» (<9) και των «άριστων» (>18)




```
ΠΑ ← 0
ΠΚ ← 0
για i από 1 μέχρι 90
  Αν Β[i] < 9 τότε
    ΠΚ ← ΠΚ + 1
  ΑλλιώςΑν (Β[i] > 18) τότε
    ΠΑ ← ΠΑ + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε ΠΚ/90*100, ΠΑ/90*100, '%'
```

Βρίσκει τον μικρότερο βαθμό των «άριστων» (>18)



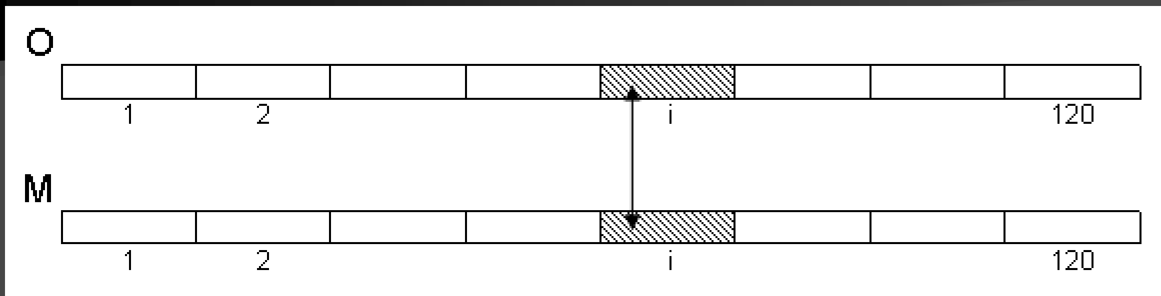

```
minA ← 21 ! κάτι μεγάλο
για i από 1 μέχρι 90
  Αν Β[i] > 18 ΚΑΙ Β[i] < minA τότε
    minA ← B[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν minA <> 21 τότε
  Γράψε minA
Αλλιώς
  Γράψε 'κανένας'
ΤέλοςΑν
```

Βρίσκει τη μικρότερη απόκλιση (απόλυτη διαφορά) δύο βαθμών

```
min ← Α_Τ(Β[1]-Β[2])
για i από 1 μέχρι 89
  για j από i+1 μέχρι 90
    d ← Α_Τ(Β[i]-Β[j])
    Αν d < min τότε
      min ← d
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε min
```

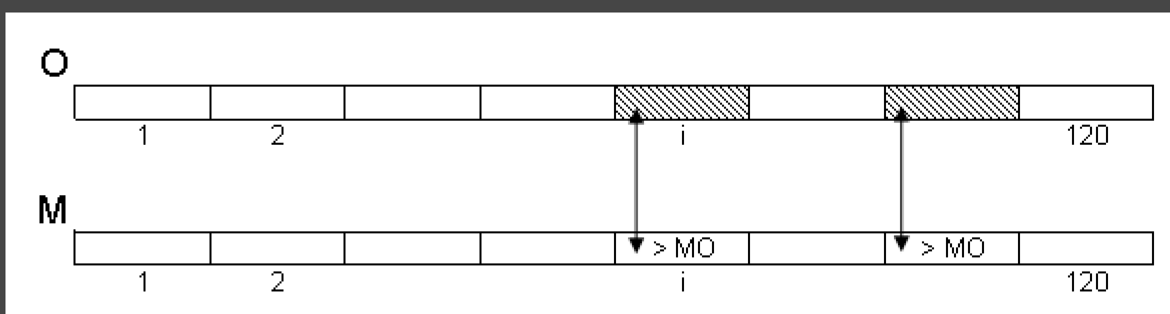
Παράλληλοι 1-Δ πίνακες

Πρόγραμμα το οποίο:
Διαβάζει σε κατάλληλους πίνακες τα ονόματα και τους μισθούς των 120 υπαλλήλων μιας εταιρείας, με έλεγχο εγκυρότητας στους μισθούς, ώστε να είναι θετικοί.



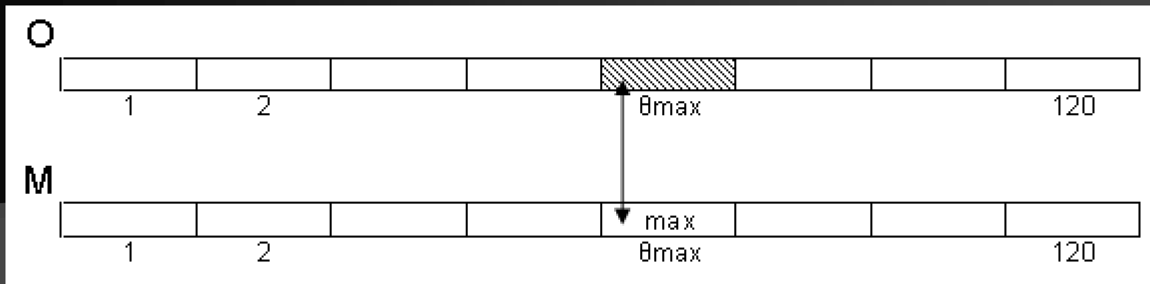

```
για i από 1 μέχρι 120
  Διάβασε Ο[i]
  ΑρχήΕπανάληψης
    Διάβασε Μ[i]
  ΜέχριςΌτου Μ[i] > 0
ΤέλοςΕπανάληψης
```

Βρίσκει ποιοι και πόσοι υπάλληλοι έχουν μισθό άνω του ΜΟ



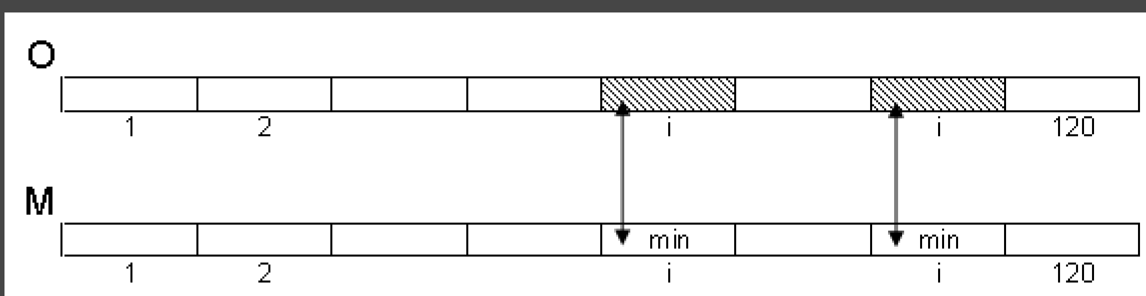

```
S ← 0
για i από 1 μέχρι 120
  S ← S + Μ[i]
ΤέλοςΕπανάληψης
ΜΟ ← S / 120
Π ← 0
για i από 1 μέχρι 120
  Αν Μ[i] > ΜΟ τότε
    Γράψε Ο[i]
    Π ← Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Π
```

Βρίσκει τον μεγαλύτερο μισθό και τον υπάλληλο που τον έχει (χωρίς ισοτιμία)



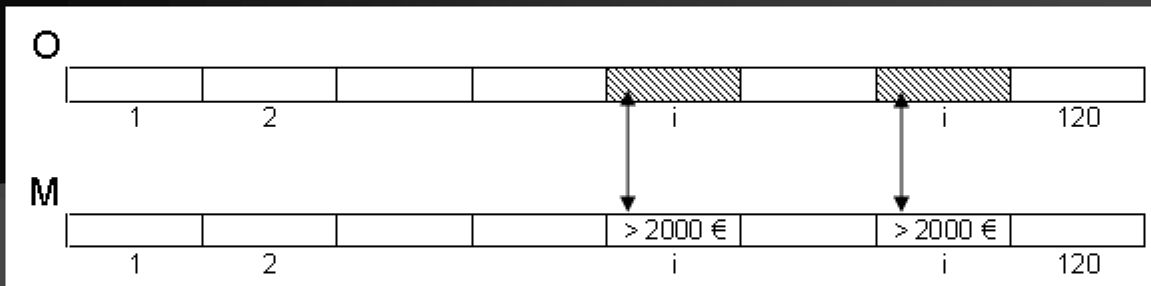

```
max ← M[1]
θmax ← 1
για i από 2 μέχρι 120
  Αν M[i] > max τότε
    max ← M[i]
    θmax ← i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε max, O[θmax]
```

Βρίσκει τον μικρότερο μισθό και τους υπαλλήλους που τον έχουν (με ισοτιμία)




```
min ← M[1]
για i από 2 μέχρι 120
  Αν M[i] < min τότε
    min ← M[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε min
για i από 1 μέχρι 120
  Αν M[i] = min τότε
    Γράψε O[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Βρίσκει το % ποσοστό των υψηλόμισθων υπαλλήλων (>2000 €) καθώς και ποιοι είναι αυτοί




```
Π ← 0
για i από 1 μέχρι 120
  Αν Μ[i] > 2000 τότε
    Π ← Π + 1
    Γράψε Ο[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Π/120*100, '%'
```

Πότε <u>απαιτείται</u> η χρήση πίνακα

Όταν χρειάζεται η αποθήκευση δεδομένων γνωστού πλήθους (προκαθορισμένου ή γνωστού μεγίστου μεγέθους) και του ίδιου τύπου, για <u>πολλαπλή</u> σάρωση και η παραμονή τους στη μνήμη μέχρι τον τερματισμό του προγράμματος. Παραδείγματα:

π.χ.1 Εισαγωγή του ύψους 100 μαθητών και εύρεση του μέσου όρου τους

! Χωρίς πίνακα

```
S ← 0
για i από 1 μέχρι 100
  Διάβασε υ
  S ← S + υ
ΤέλοςΕπανάληψης
ΜΟ ← S / 100
Γράψε ΜΟ
```

! Με πίνακα

```
S ← 0
για i από 1 μέχρι 100
  Διάβασε Υ[i]
  S ← S + Υ[i]
ΤέλοςΕπανάληψης
ΜΟ ← S / 100
Γράψε ΜΟ
```

Συμπέρασμα:
δεν απαιτείται η χρήση πίνακα

Πότε <u>απαιτείται</u> η χρήση πίνακα

π.χ.2 Εισαγωγή του ύψους 100 μαθητών και εύρεση του πλήθους των μαθητών με ύψος άνω του ΜΟ

π.χ.3 Εισαγωγή 100 τιμών και εμφάνισή τους με την ανάποδη σειρά εισαγωγής.

! Χωρίς πίνακα

```
S ← 0
για i από 1 μέχρι 100
  Διάβασε υ
  S ← S + υ
ΤέλοςΕπανάληψης
ΜΟ ← S / 100
Π ← 0
για i από 1 μέχρι 100
  Αν υ > ΜΟ τότε ! υ ??????
     Π ← Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Π
```

! Με πίνακα

```
S ← 0
για i από 1 μέχρι 100 ! 1η σάρωση
  Διάβασε Y[i]
  S ← S + Y[i]
ΤέλοςΕπανάληψης
ΜΟ ← S / 100
Π ← 0
για i από 1 μέχρι 100 ! 2η σάρωση
  Αν Y[i] > ΜΟ τότε
     Π ← Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Π
```

Συμπέρασμα:
<u>απαιτείται</u> η χρήση πίνακα

Μειονεκτήματα χρήσης πινάκων:

1. Απαιτούν μνήμη
2. Περιορίζουν τις δυνατότητες του προγράμματος

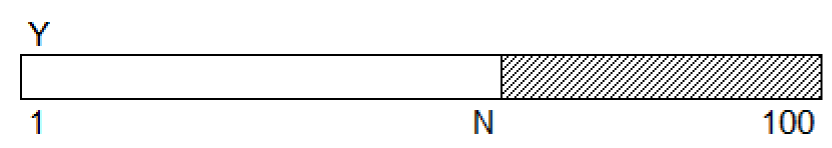
Πότε μπορεί να γίνει χρήση πίνακα

Όταν γίνεται διαχείριση ενός συνόλου δεδομένων του ίδιου τύπου

- γνωστού και προκαθορισμένου πλήθους
- ή γνωστού και προκαθορισμένου μεγίστου πλήθους

π.χ Εισαγωγή του ύψους μαθητών μέχρι να γίνουν το πολύ 100 ή να δοθεί ως ύψος η τιμή -1. Εύρεση του πλήθους των μαθητών με ύψος μεγαλύτερο του μέσου όρου τους.

```
S ← 0 N ← 0
Διάβασε υ
Όσο υ <> -1 ΚΑΙ N < 100 επανάλαβε
  N ← N + 1
  Y[N] ← υ
  S ← S + Y[N]
   Αν N < 100 τότε
    Διάβασε υ
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν N = 0 τότε
  Γράψε ‘Κανένας μαθητής
Αλλιώς
  Π ← 0 ΜΟ ← S / N
  για i από 1 μέχρι N
    Αν Y[i] > ΜΟ τότε
       Π ← Π + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Γράψε Π
ΤέλοςΑν
```



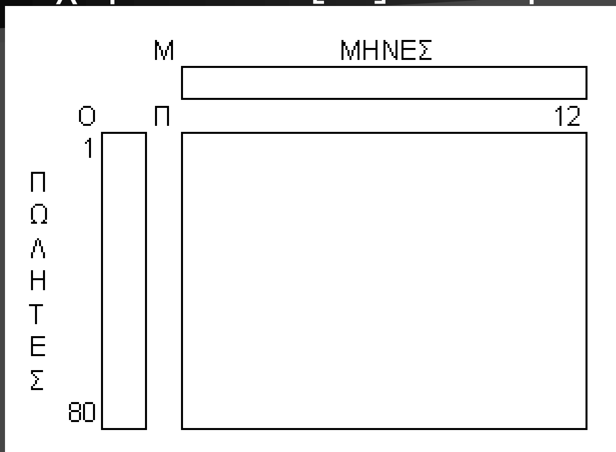
Επεξεργασία 2-Δ πίνακα

Πρόγραμμα το οποίο:

Περιλαμβάνει τμήμα δήλωσης μεταβλητών. Διαβάζει σε κατάλληλους πίνακες:

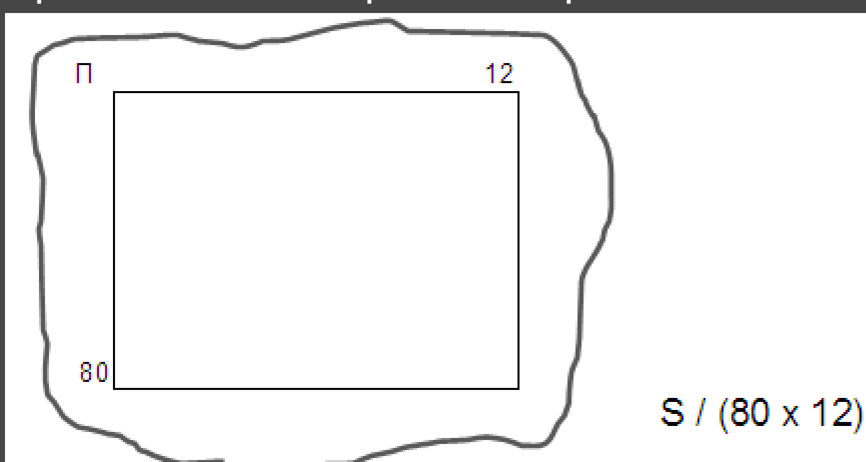
- τα ονόματα των 80 πωλητών μιας εταιρείας
- τις μηνιαίες πωλήσεις τους (€) για τους 12 μήνες ενός έτους με έλεγχο εγκυρότητας ώστε να είναι θετικές (≥ 0)

Καταχωρεί στον Μ[12] τα ονόματα των 12 μηνών



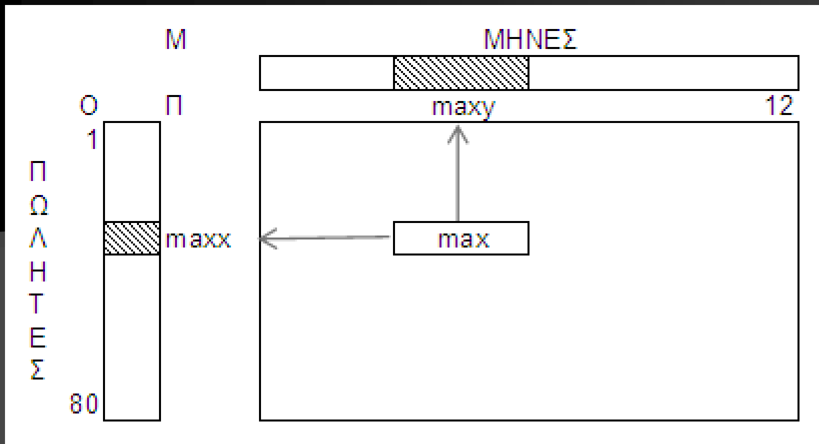

```
Πρόγραμμα Εταιρεία
  Μεταβλητές
    Χαρακτήρες: Ο[80], Μ[12]
    Πραγματικές: Π[80, 12]
    Ακέραιες: i, j
Αρχή
  Μ[1] <-- 'Ιανουάριος'
  ...
  Μ[12] <-- 'Δεκέμβριος'
για i από 1 μέχρι 80
  Διάβασε Ο[i]
  για j από 1 μέχρι 12
    ΑρχήΕπανάληψης
      Διάβασε Π[i, j]
    ΜέχριςΌτου Π[i, j] >= 0
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Βρίσκει τον ΜΟ ετησίων πωλήσεων όλων των πωλητών



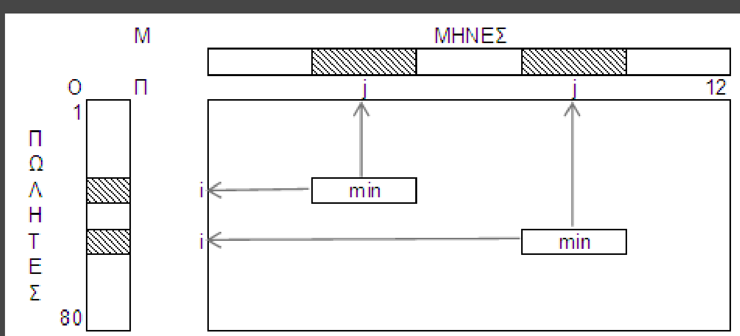

```
S ← 0
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    S ← S + Π[i,j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
ΜΟ ← S  /(80*12)
Γράψε ΜΟ
```

Βρίσκει τη μέγιστη μηνιαία πώληση, ποιός την έκανε και σε ποιό μήνα (χωρίς ισοτιμία)




```
max ← Π[1,1]
maxx ← 1
maxy ← 1
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    Αν Π[i,j] > max τότε
      max ← Π[i,j]
      maxx ← i
      maxy ← j
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε max, Ο[maxx], Μ[maxy]
```

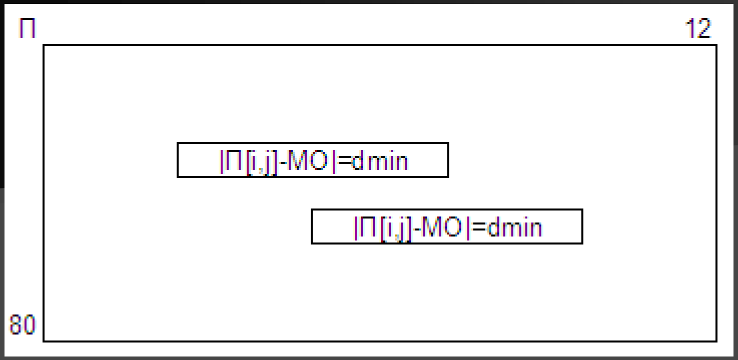
Βρίσκει τη μικρότερη μηνιαία πώληση, ποιοί την έκαναν και σε ποιούς μήνες (με ισοτιμία)




```
min ← Π[1,1]
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    Αν Π[i,j] < min τότε
      min ← Π[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε min
```

```
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    Αν Π[i,j] = min τότε
      Γράψε Ο[i], Μ[j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

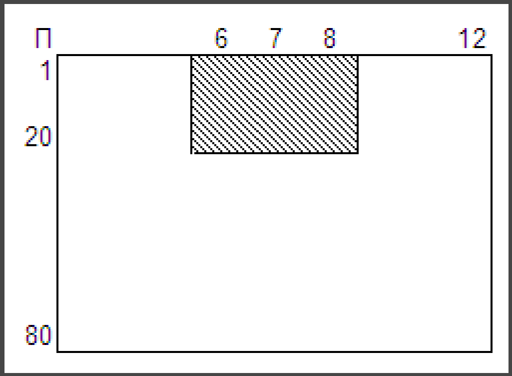
Βρίσκει πόσες πωλήσεις ήταν πλησιέστερες προς το ΜΟ



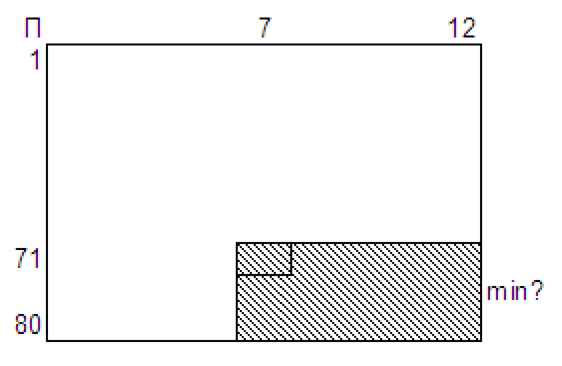
Βρίσκει τη μέση πώληση των 20 πρώτων πωλητών κατά την καλοκαιρινή περίοδο

```
S ← 0
για i από 1 μέχρι 20
  για j από 6 μέχρι 8
    S ← S + Π[i,j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
ΜΟ ← S  /(20*3)
Γράψε ΜΟ
```

```
! ο ΜΟ έχει υπολογισθεί
dmin ← Α_Τ(Π[1,1] – ΜΟ)
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    Αν Α_Τ(Π[i,j] – ΜΟ)  < dmin τότε
      dmin ← Α_Τ(Π[i,j] – ΜΟ)
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Π1 ← 0
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    Αν Α_Τ(Π[i,j] – ΜΟ) = dmin τότε
      Π1 ← Π1 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε Π1
```

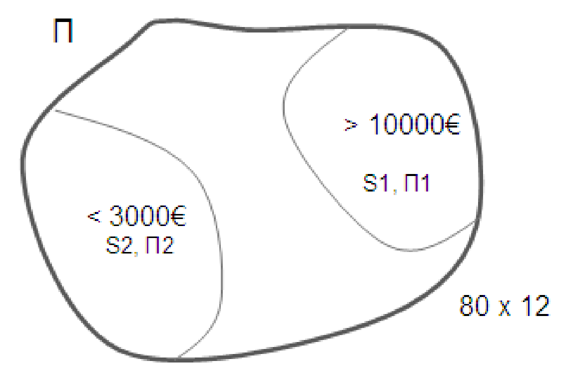


Βρίσκει την ελάχιστη πώληση των 10 τελευταίων πωλητών κατά το 2ο 6μηνο




```
min ← Π[71,7]
για i από 71 μέχρι 80
  για j από 7 μέχρι 12
    Αν Π[i,j] < min τότε
      min ← Π[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε min
```

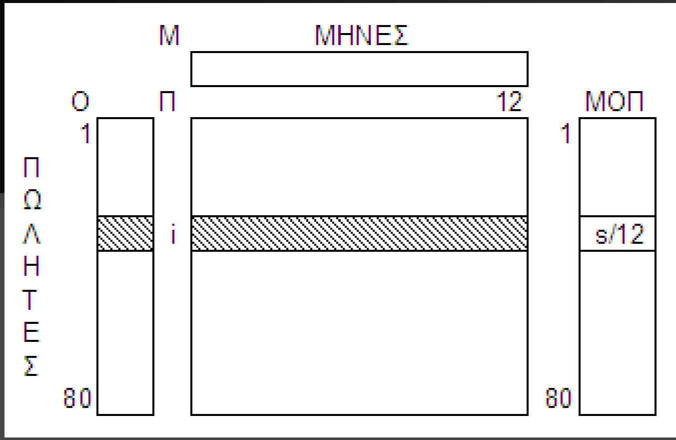
Βρίσκει τους ΜΟ των πωλήσεων > 10000€ και των πωλήσεων < 3000€




```
S1 ← 0
Π1 ← 0
S2 ← 0
Π2 ← 0
για i από 1 μέχρι 80
  για j από 1 μέχρι 12
    Αν Π[i,j] > 10000 τότε
      S1 ← S1 + Π[i,j]
      Π1 ← Π1 + 1
    ΑλλιώςΑν Π[i,j] < 3000 τότε
      S2 ← S2 + Π[i,j]
      Π2 ← Π2 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

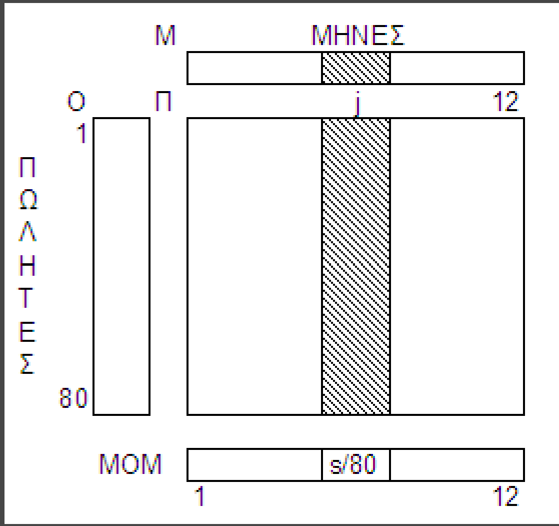
```
Αν Π1<> 0 τότε
  Γράψε S1/Π1
Αλλιώς
  Γράψε ‘Καμία πώληση > 10000’
ΤέλοςΑν
Αν Π2<> 0 τότε
  Γράψε S2/Π2
Αλλιώς
  Γράψε ‘Καμία πώληση < 3000’
ΤέλοςΑν
```

Εμφανίζει τον κάθε πωλητή με το ΜΟ των πωλήσεών του



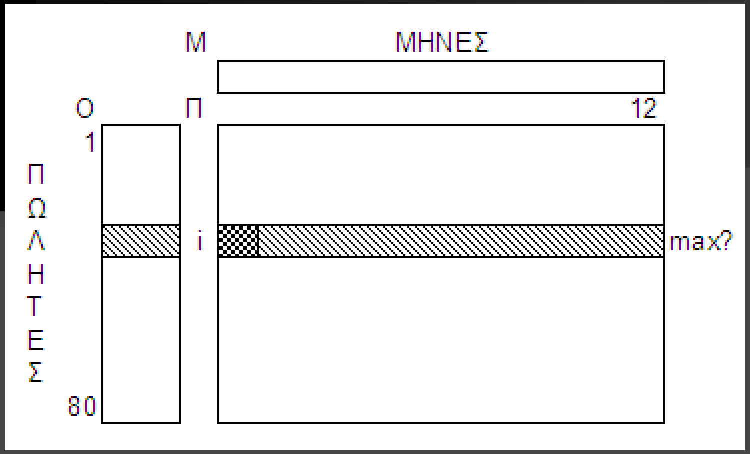

```
για i από 1 μέχρι 80
  s ← 0
  για j από 1 μέχρι 12
    s ← s + Π[i,j]
  ΤέλοςΕπανάληψης
  ΜΟΠ[i] ← s / 12
  Γράψε Ο[i], ΜΟΠ[i]
ΤέλοςΕπανάληψης
```

Εμφανίζει τον κάθε μήνα με το ΜΟ των πωλήσεών του



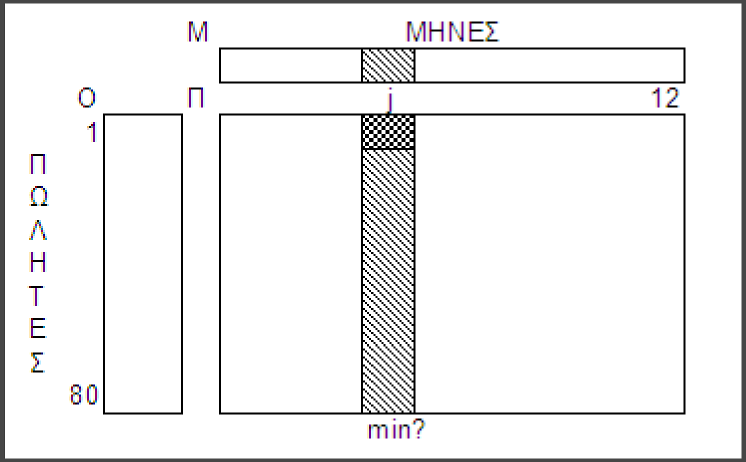

```
για j από 1 μέχρι 12
  s ← 0
  για i από 1 μέχρι 80
    s ← s + Π[i,j]
  ΤέλοςΕπανάληψης
  ΜΟΜ[j] ← s / 80
  Γράψε Μ[j], ΜΟΜ[j]
ΤέλοςΕπανάληψης
```

Εμφανίζει τον κάθε πωλητή με την υψηλότερη μηνιαία πώλησή του



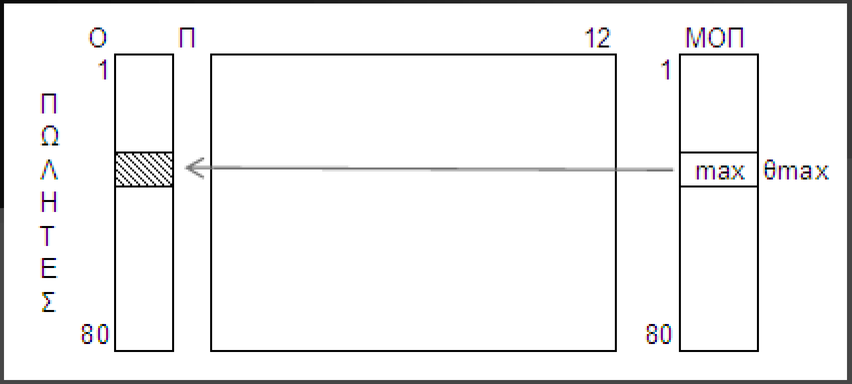

```
για i από 1 μέχρι 80
  max ← Π[i, 1]
  για j από 2 μέχρι 12
    Αν Π[i,j] > max τότε
      max ← Π[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Γράψε Ο[i], max
ΤέλοςΕπανάληψης
```

Εμφανίζει τον κάθε μήνα με τη χαμηλότερη μηνιαία πώλησή του



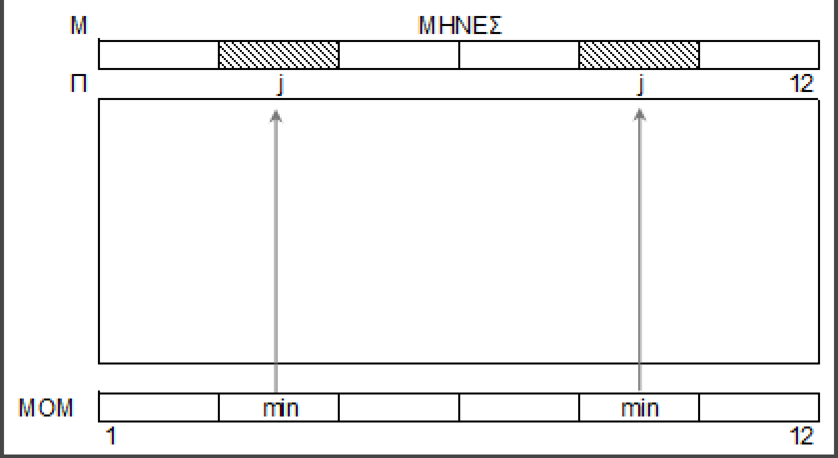

```
για j από 1 μέχρι 12
  min ← Π[1, j]
  για i από 2 μέχρι 80
    Αν Π[i,j] < min τότε
      min ← Π[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Γράψε Μ[j], min
ΤέλοςΕπανάληψης
```

Βρίσκει τον «πωλητή της χρονιάς» (μόνο ένας)



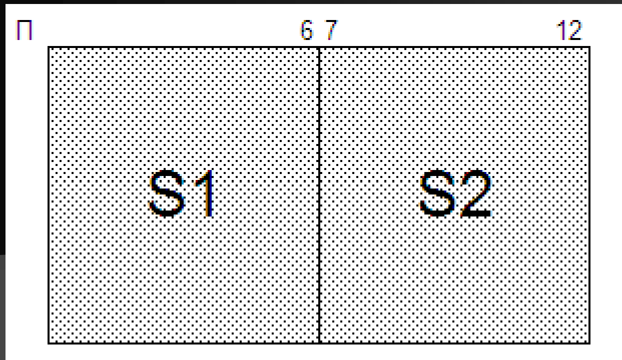
max ← ΜΟΠ[1]
θmax ← 1
για i από 2 μέχρι 80
  Αν ΜΟΠ[i] > max τότε
    max ← ΜΟΠ[i]
    θmax ← i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Ο[θmax]

Βρίσκει τους «χειρότερους» μήνες του έτους (με ισοτιμία)



min ← ΜΟΜ[1]
για j από 2 μέχρι 12
  Αν ΜΟΜ[j] < min τότε
    min ← ΜΟΜ[j]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για j από 1 μέχρι 12
  Αν ΜΟΜ[j] = min τότε
    Γράψε Μ[j]
  ΤέλοςΑν
ΤέλοςΕπανάληψης

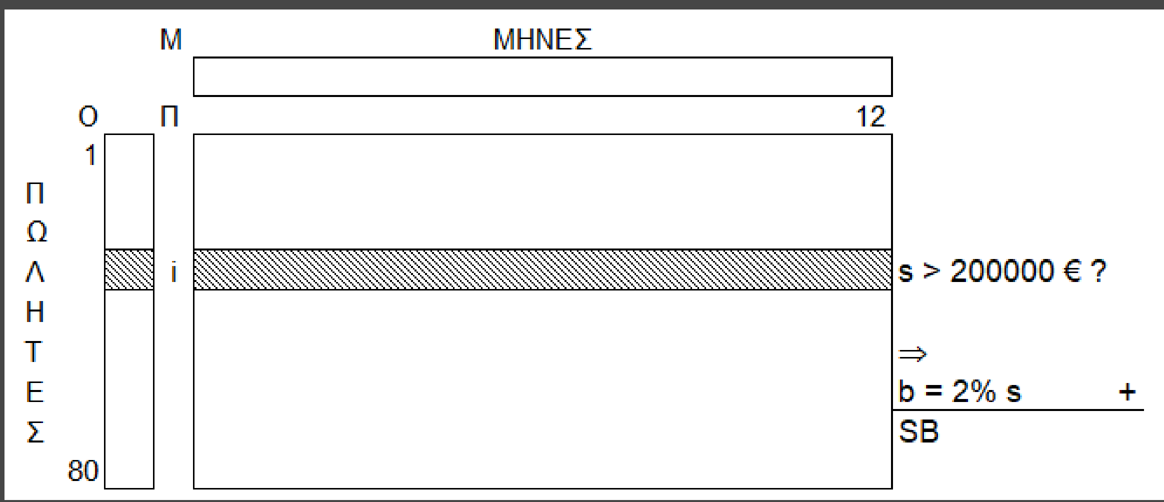
Ποιό ήταν το ποιό αποδοτικό 6-μηνο;




```
S1 ← 0
S2 ← 0
για i από 1 μέχρι 80
  για j από 1 μέχρι 6
    S1 ← S1 + Π[i, j]
    S2 ← S2 + Π[i, j + 6]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

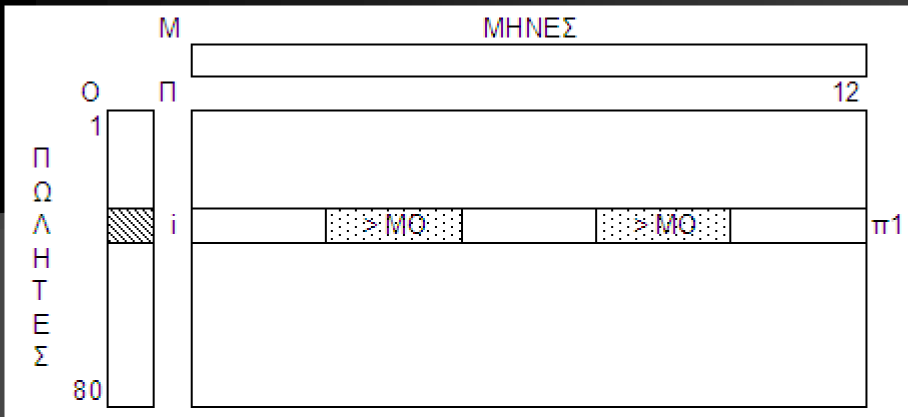
```
Αν S1 > S2 τότε
  Γράψε ‘1ο’
ΑλλιώςΑν S2 > S1 τότε
  Γράψε ‘2ο’
Αλλιώς
  Γράψε ‘ισοπαλία’
ΤέλοςΑν
```

Έστω ότι δίνεται bonus 2% επί των ετησίων πωλήσεων του κάθε πωλητή, στους πωλητές με ετήσιες πωλήσεις > 200000€. Ποιοί και πόσοι πωλητές το παίρνουν, πόσο παίρνουν και πόσο το σύνολο του bonus;



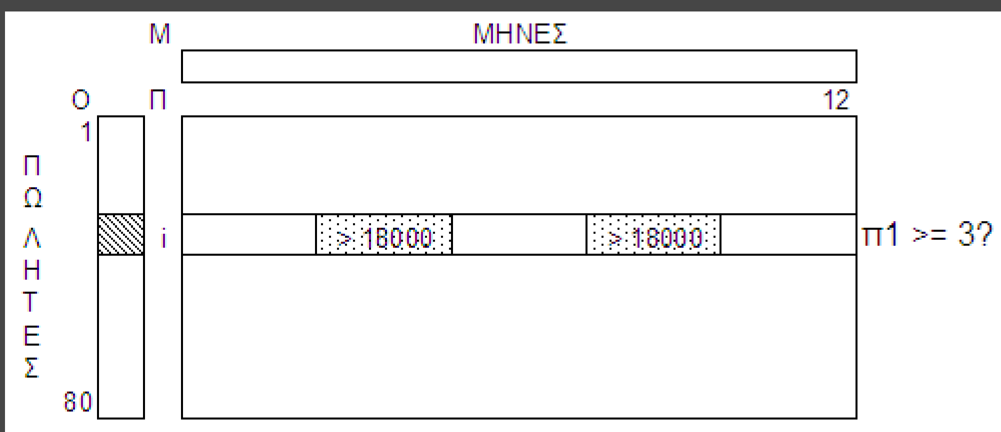

```
Π1 ← 0
SB ← 0
για i από 1 μέχρι 80
  s ← 0
  για j από 1 μέχρι 12
    s ← s + Π[i,j]
  ΤέλοςΕπανάληψης
  Αν s > 200000 τότε
    b ← 2/100 * s
    Γράψε Ο[i], b
    Π1 ← Π1 + 1
    SB ← SB + b
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Π1, SB
```

Σε πόσους μήνες κάθε πωλητής είχε πωλήσεις > ΜΟ της εταιρείας;



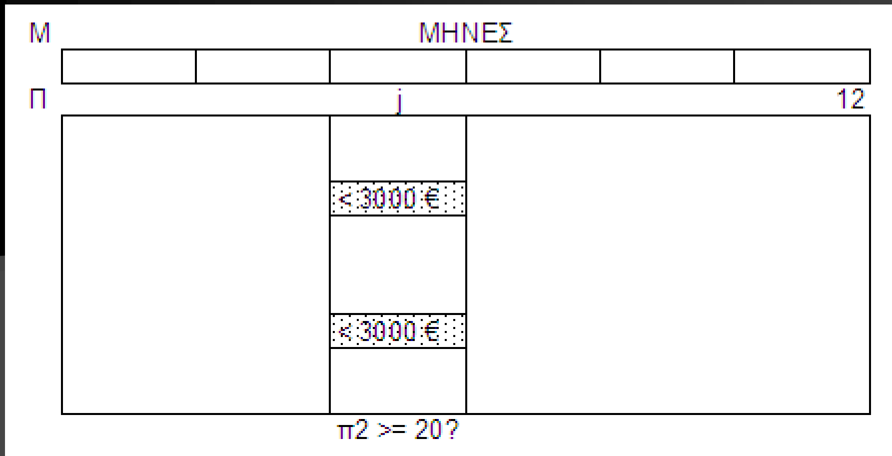

```
! ο ΜΟ έχει υπολογισθεί
για i από 1 μέχρι 80
  π1 ← 0
  για j από 1 μέχρι 12
    Αν Π[i,j] > ΜΟ τότε
      π1 ← π1 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Γράψε Ο[i], π1
ΤέλοςΕπανάληψης
```

Ποιοί πωλητές είχαν 3 τουλάχιστον πώλησεις > 18000€;




```
για i από 1 μέχρι 80
  π1 ← 0
  για j από 1 μέχρι 12
    Αν Π[i,j] > 18000 τότε
      π1 ← π1 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν π1 >= 3 τότε
    Γράψε Ο[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Σε ποιούς και πόσους μήνες έγιναν 20 τουλάχιστον πωλήσεις < 3000€;



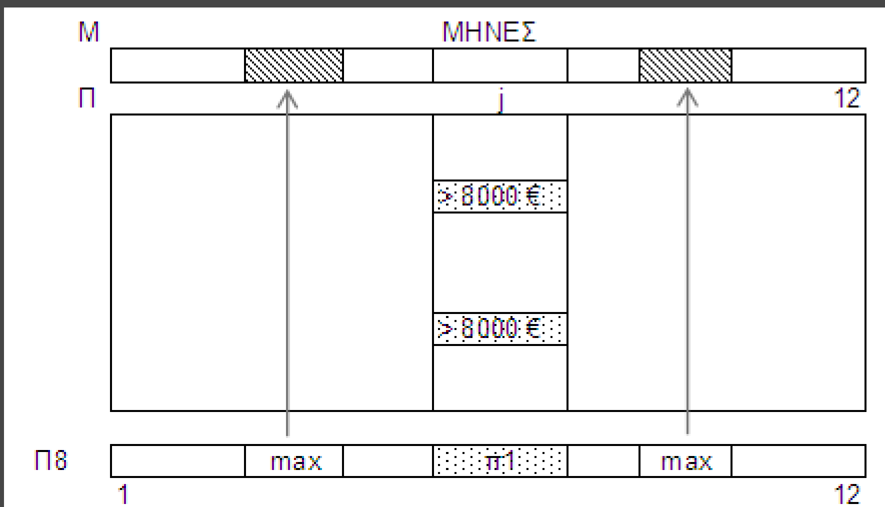

```
π1 ← 0
για j από 1 μέχρι 12
  π2 ← 0
  για i από 1 μέχρι 80
    Αν Π[i,j] < 3000 τότε
      π2 ← π2 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν π2 >= 20 τότε
    Γράψε Μ[j]
    π1 ← π1 + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε π1
```

Ποιοί πωλητές είχαν το 2ο μεγαλύτερο μέσο όρο πωλήσεων;

```
max1 ← -1 max2 ← -1
για i από 1 μέχρι 80
  Αν ΜΟΠ[i] > max1 τότε
    max2 ← max1
    max1 ← ΜΟΠ[i]
  ΑλλιώςΑν ΜΟΠ[i] > max2 τότε
    max2 ← ΜΟΠ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

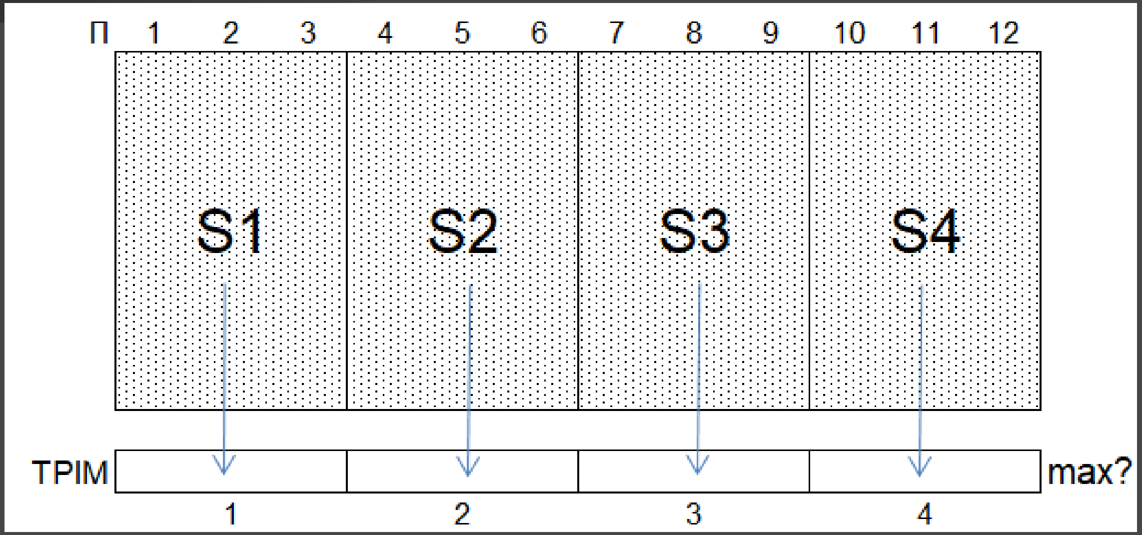
```
για i από 1 μέχρι 80
    Αν ΜΟΠ[i] = max2 τότε
       Γράψε Ο[i]
    ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Ποιοί μήνες είχαν τις περισσότερες πωλήσεις > 8000€;



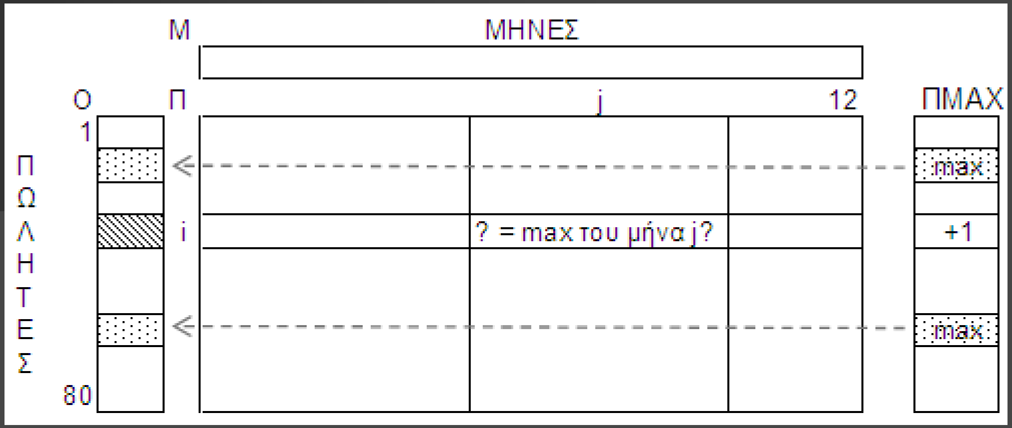

```
για j από 1 μέχρι 12
  π1 ← 0
  για i από 1 μέχρι 80
    Αν Π[i,j] > 8000 τότε
      π1 ← π1 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Π8[j] ← π1
ΤέλοςΕπανάληψης
max ← Π8[1]
για i από 2 μέχρι 12
  Αν Π8[i] > max τότε
    max ← Π8[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 12
  Αν Π8[i] = max τότε
    Γράψε Μ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
για i από 1 μέχρι 4
    ΤΡΙΜ[i] ← 0
  ΤέλοςΕπανάληψης
για j από 1 μέχρι 12
  τρ ← j div 3
  Αν j mod 3 <> 0 τότε
    τρ ← τρ + 1
 ΤέλοςΑν
  για i από 1 μέχρι 80
    ΤΡΙΜ[τρ] ← ΤΡΙΜ[τρ] + Π[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
max ← ΤΡΙΜ[1]
για i από 2 μέχρι 4
  Αν ΤΡΙΜ[i] > max τότε
    max ← ΤΡΙΜ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 4
  Αν ΤΡΙΜ[i] = max τότε
    Γράψε i, 'ο 3μηνο'
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```



Ποιο 3μηνο ήταν το πιο αποδοτικό;

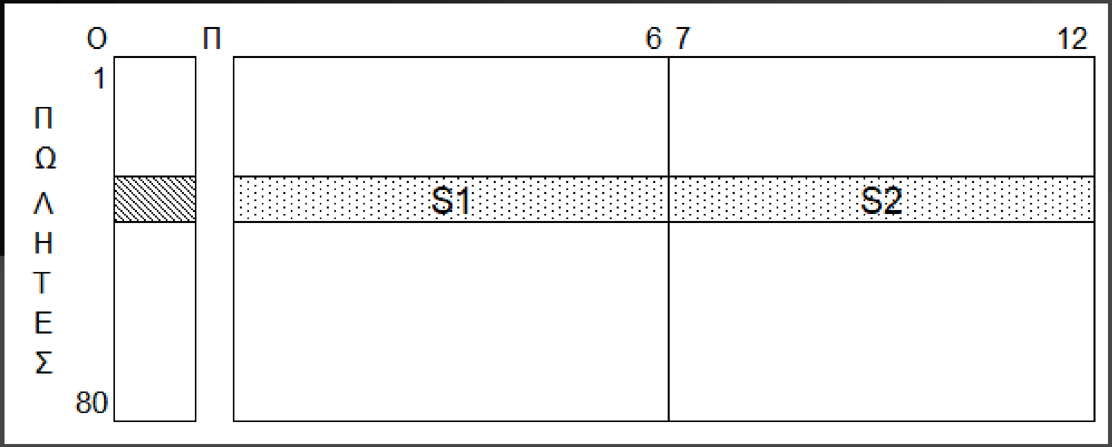
Ποιοι πωλητές ήταν οι καλύτεροι τους περισσότερους μήνες;




```
για i από 1 μέχρι 80
  ΠΜΑΧ[i] ← 0
τέλος_επανάληψης
για j από 1 μέχρι 12
  max ← Π[1, j]
  για i από 2 μέχρι 80
    Αν Π[i, j] > max τότε
      max ← Π[i,j]
    ΤέλοςΑν
  τέλος_επανάληψης
  για i από 1 μέχρι 80
    Αν Π[i, j] = max τότε
      ΠΜΑΧ[i] ← ΠΜΑΧ[i] + 1
    ΤέλοςΑν
  τέλος_επανάληψης
τέλος_επανάληψης
```

```
! max με ισοτιμία στον ΠΜΑΧ[80]
max ← ΠΜΑΧ[1]
για i από 2 μέχρι 80
  Αν ΠΜΑΧ[i] > max τότε
    max ← ΠΜΑΧ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 80
  Αν ΠΜΑΧ[i] = max τότε
    Γράψε Ο[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

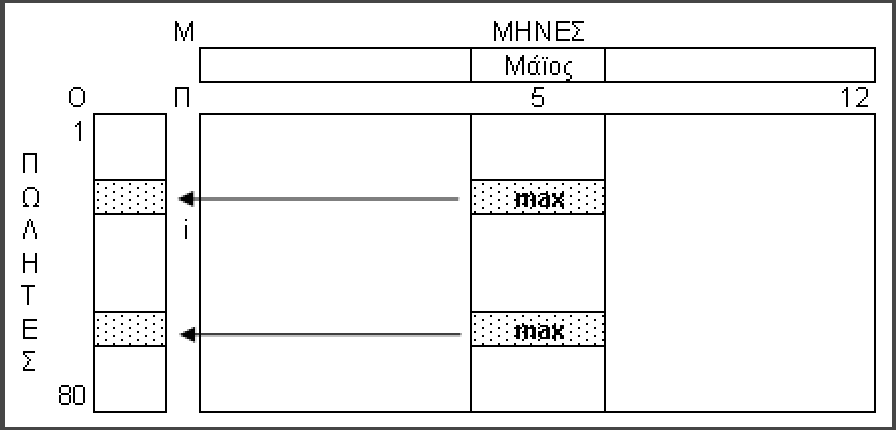
Ποιοί και πόσοι πωλητές είχαν στο 2ο 6μηνο σύνολο πωλήσεων μεγαλύτερο από του 1ου;




```
Π1 ← 0
για i από 1 μέχρι 80
  S1 ← 0
  S2 ← 0
  για j από 1 μέχρι 6
    S1 ← S1 + Π[i,j]
    S2 ← S2 + Π[i,j+6]
  ΤέλοςΕπανάληψης
  Αν S2 > S1 τότε
    Γράψε Ο[i]
    Π1 ← Π1 + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε Π1
```

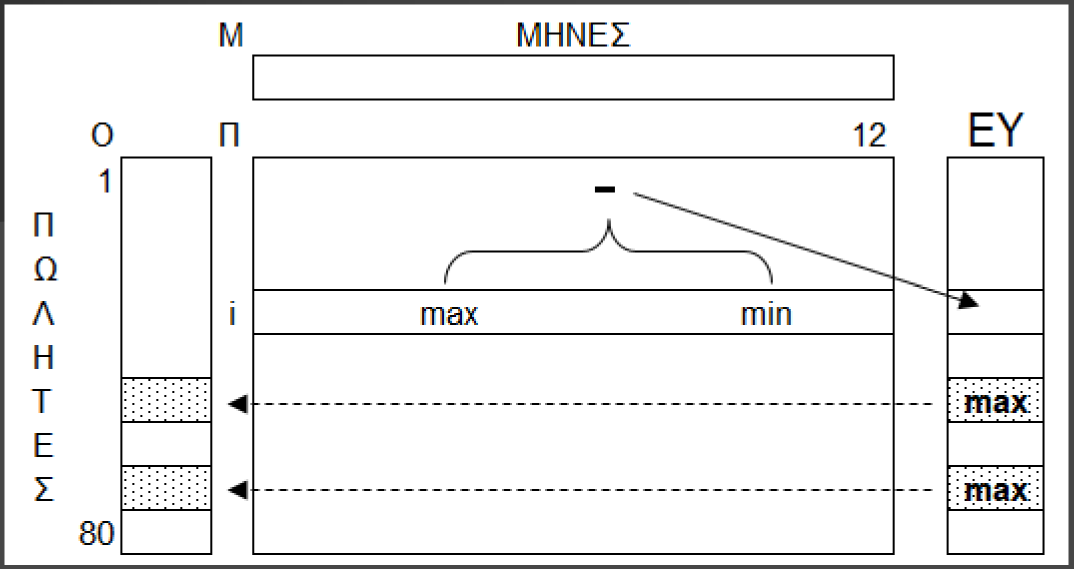
Ποιοι πωλητές είχαν τις υψηλότερες πωλήσεις τον 5ο μήνα;

```
max ← Π[1, 5]
για i από 2 μέχρι 80
  Αν Π[i, 5] > max τότε
    max ← Π[i, 5]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 80
  Αν Π[i, 5] = max τότε
    Γράψε Ο[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```



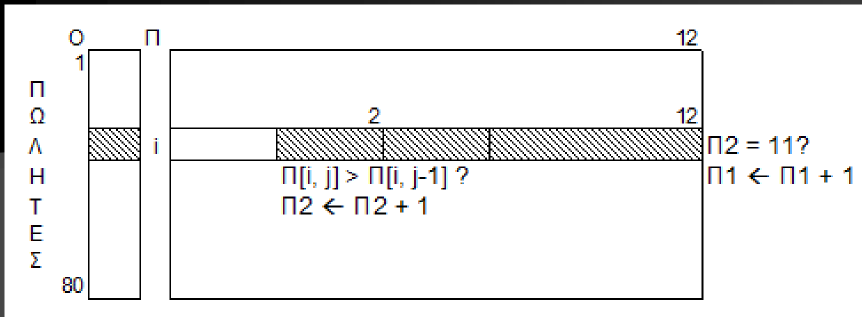
Επεξεργασία 2-Δ πίνακα

```
για i από 1 μέχρι 80
  max ← Π[i, 1]
  min ← Π[i, 1]
  για j από 2 μέχρι 12
    Αν Π[i,j] > max τότε
      max ← Π[i,j]
    ΤέλοςΑν
    Αν Π[i,j] < min τότε
      min ← Π[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  ΕΥ[i] ← max - min
ΤέλοςΕπανάληψης
max ← ΕΥ[1]
για i από 2 μέχρι 80
  Αν ΕΥ[i] > max τότε
    max ← ΕΥ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 80
  Αν ΕΥ[i] = max τότε
    Γράψε Ο[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```



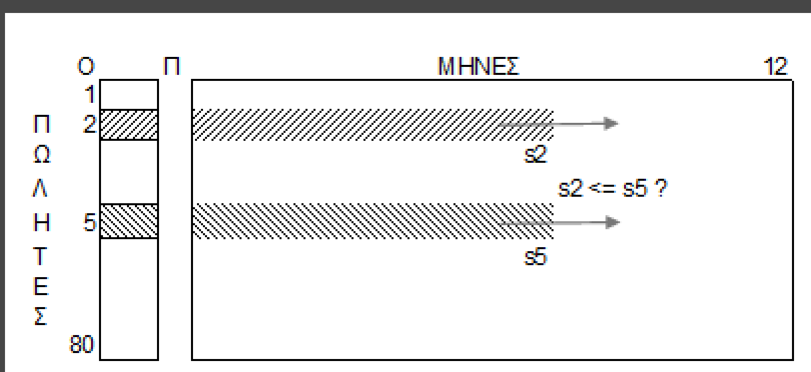
Ποιοί πωλητές είχαν το μεγαλύτερο «εύρος» (διαφορά μέγιστης και ελάχιστης πώλησης) πωλήσεων ;

Ποιοι πωλητές είχαν συνεχόμενη αύξηση μηνιαίων πωλήσεων (Φεβ-Δεκ). Αν δεν υπήρξαν, να εμφανίζει κατάλληλο μήνυμα




```
Π1 ← 0
για i από 1 μέχρι 80
  Π2 ← 0
  για j από 2 μέχρι 12
    Αν Π[i, j] > Π[i, j-1] τότε
      Π2 ← Π2 + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν Π2 = 11 τότε
    Γράψε Ο[i]
    Π1 ← Π1 + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν Π1 = 0 τότε
  Γράψε 'Κανένας'
ΤέλοςΑν
```

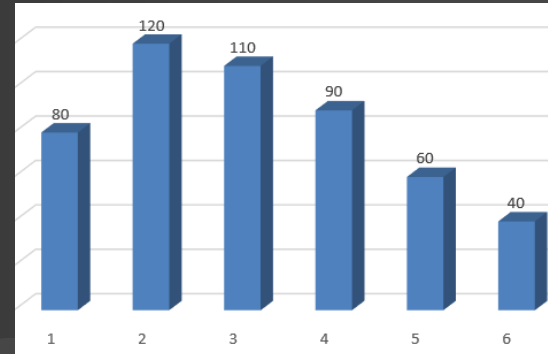
Σε πόσους μήνες κατάφερε ο 2^{ος} πωλητής να έχει για πρώτη φορά σύνολο πωλήσεων μεγαλύτερο από τον 5^{ο}; Αν δεν το κατάφερε ποτέ, να εμφανίζει κατάλληλο μήνυμα

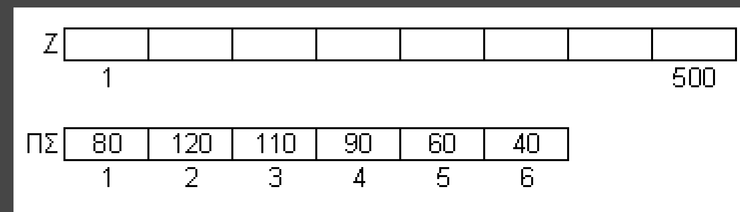



```
s2 ← 0 s5 ← 0
j ← 0
Όσο j < 12 ΚΑΙ s2 <= s5 επανάλαβε
  j ← j + 1
  s2 ← s2 + Π[2,j]
  s5 ← s5 + Π[5,j]
ΤέλοςΕπανάληψης
Αν s2 > s5 τότε
  Γράψε 'Σε ', j, ' μήνες'
Αλλιώς
  Γράψε 'ποτέ'
ΤέλοςΑν
```

Πίνακας συχνοτήτων

π.χ.1 Δίνεται ο Z[500] με τις τυχαίες ενδείξεις των 500 ρίψεων ενός ζαριού (1-6). Πρόγραμμα που εμφανίζει τα % ποσοστά εμφάνισης της κάθε ένδειξης (1-6). Ποια ένδειξη ήταν η πιο συχνή; (με ισοτιμία)

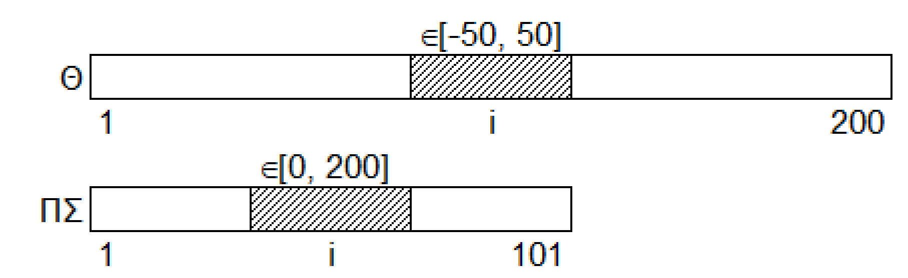





```
για i από 1 μέχρι 6
  ΠΣ[i] ← 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 500
  ΠΣ[Z[i]] ← ΠΣ[Z[i]] + 1
  ! εναλλακτικά:
  ! x ← Z[i]
  ! ΠΣ[x] ← ΠΣ[x] + 1
ΤέλοςΕπανάληψης
για i από 1 μέχρι 6
  Γράψε 'Ενδειξη: ', i, ΠΣ[i] / 500 * 100, '%'
ΤέλοςΕπανάληψης
```

```
max ← ΠΣ[1]
για i από 2 μέχρι 6
  Αν ΠΣ[i] > max τότε
    max ← ΠΣ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 6
  Αν ΠΣ[i] = max τότε
    Γράψε i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Πίνακας συχνοτήτων



π.χ.2 Δίνεται ο Θ[200] με τις ακέραιες θερμοκρασίες στο διάστημα [-50,50], 200 περιοχών στις 12:00 το μεσημέρι. Πρόγραμμα που εμφανίζει τα % ποσοστά εμφάνισης της κάθε θερμοκρασίας (-50,50). Ποια θερμοκρασία ήταν η πιο συχνή; (με ισοτιμία)

```
για i από 1 μέχρι 101
  ΠΣ[i] <-- 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 200
  ΠΣ[Θ[i]+51] <-- ΠΣ[Θ[i]+51] + 1
  ! εναλλακτικά:
  ! x <-- Θ[i]+51
  ! ΠΣ[x] <-- ΠΣ[x] + 1
ΤέλοςΕπανάληψης
για i από 1 μέχρι 101
  Γράψε 'Θερμοκρασία: ', i-51, ΠΣ[i]/200*100, '%'
ΤέλοςΕπανάληψης
```

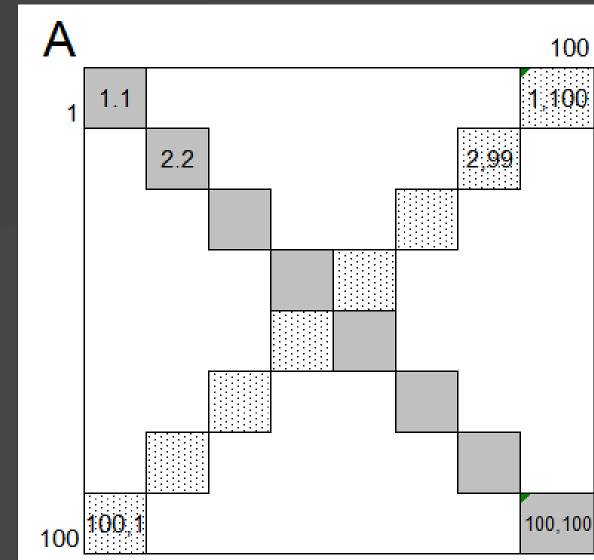
```
max <-- ΠΣ[1]
για i από 2 μέχρι 101
  Αν ΠΣ[i] > max τότε
    max <-- ΠΣ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 101
  Αν ΠΣ[i] = max τότε
    Γράψε i-51
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Τετραγωνικοί πίνακες π.χ. A[100, 100]

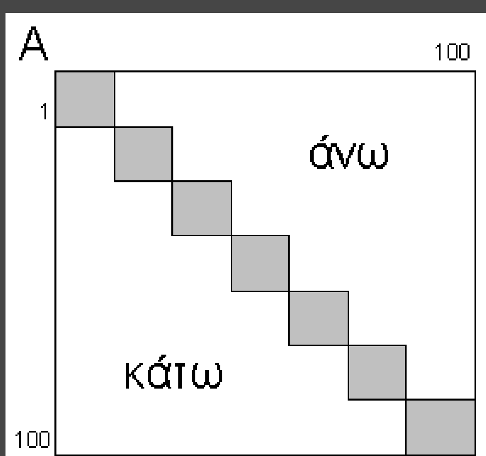
```
! 1η διαγώνιος
για i από 1 μέχρι 100
  ... A[i, i]
ΤέλοςΕπανάληψης
ή
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i=j τότε
     ... A[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

```
! 2η διαγώνιος
για i από 1 μέχρι 100
  ... A[i, 101-i]
ΤέλοςΕπανάληψης
ή
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i+j=101 τότε
     ... A[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

1η και 2η κύρια διαγώνιος:



Άνω και κάτω της 1ης κύριας διαγωνίου:




```
! Άνω
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i<j τότε
     ... A[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
ή
για i από 1 μέχρι 100
  για j από i+1 μέχρι 100
    ... A[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

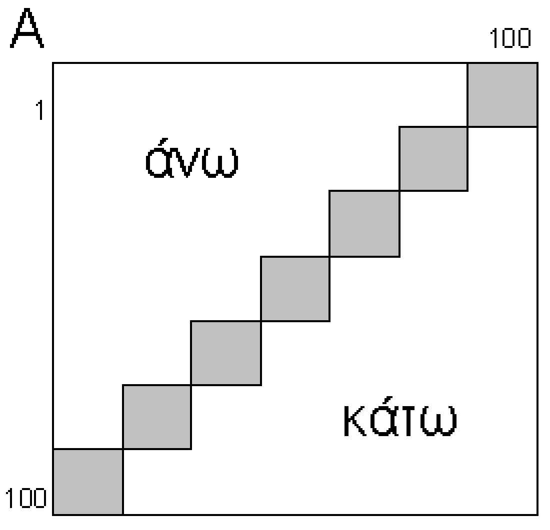
```
! Κάτω
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i>j τότε
     ... A[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
ή
για i από 1 μέχρι 100
  για j από 1 μέχρι i-1
    ... A[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

$$\frac{N^2 - N}{2}$$

Άνω και κάτω της 2ης κύριας διαγωνίου:

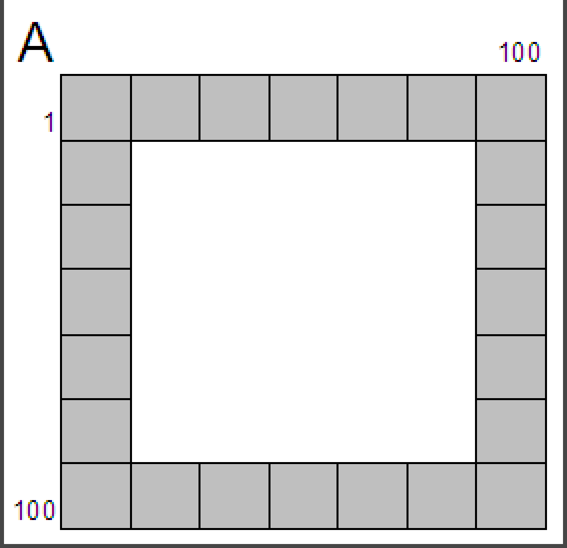
```
! Άνω
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i+j<101 τότε
     ... A[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

```
! Κάτω
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i+j>101 τότε
     ... A[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

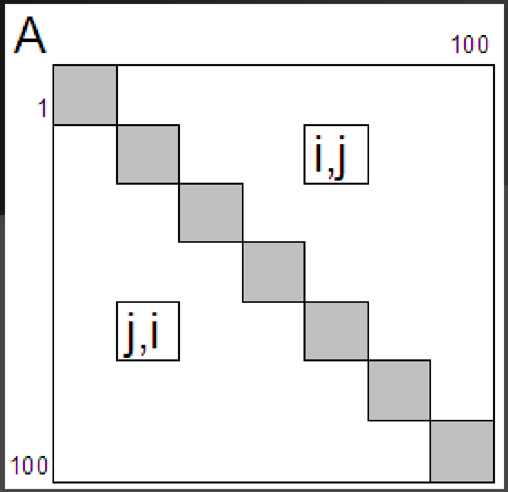


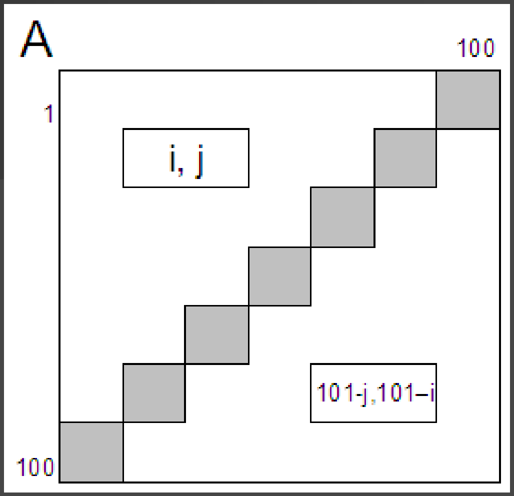
Περιφέρεια:

```
για i από 1 μέχρι 100
  ... A[1, i] ! Επάνω πλευρά
 ... A[100, i] ! Κάτω πλευρά
 ... A[i, 1] ! Αριστερή πλευρά
 ... A[i, 100] ! Δεξιά πλευρά
ΤέλοςΕπανάληψης
```

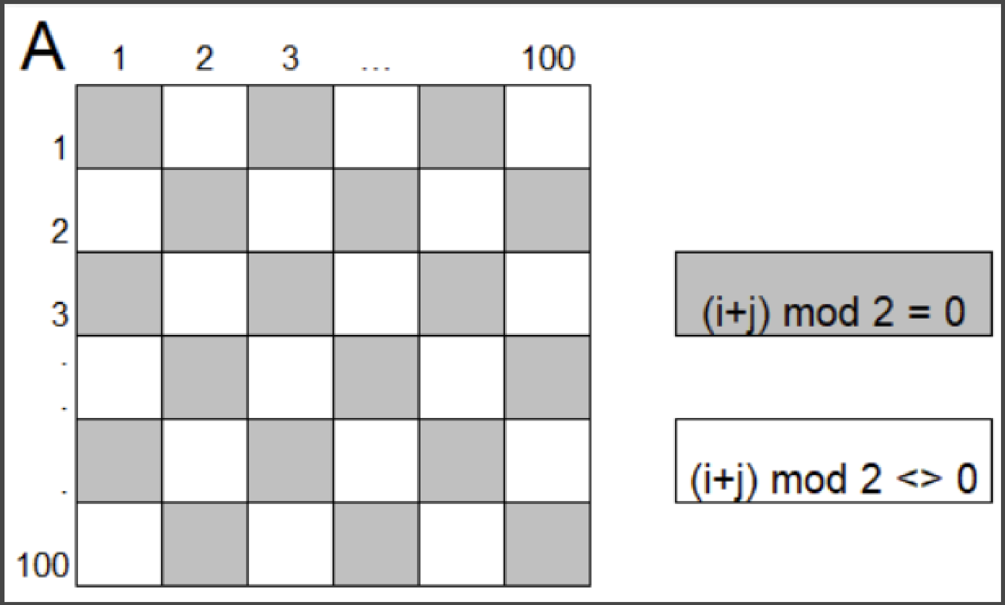


Συμμετρικά κελιά ως προς την 1η διαγώνιο και τη 2η διαγώνιο:

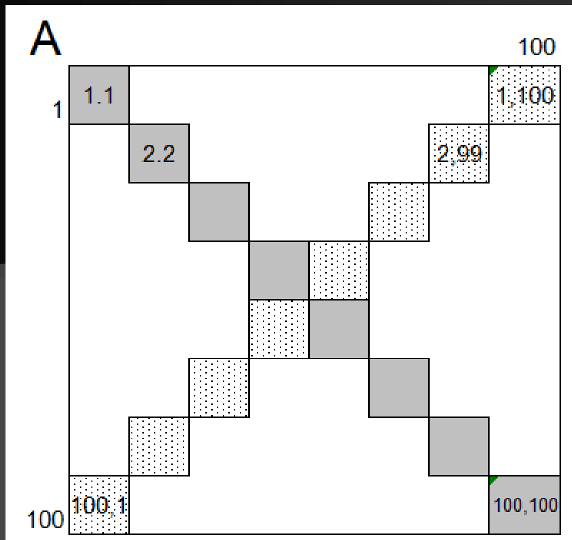




Εναλλάξ κελιά:

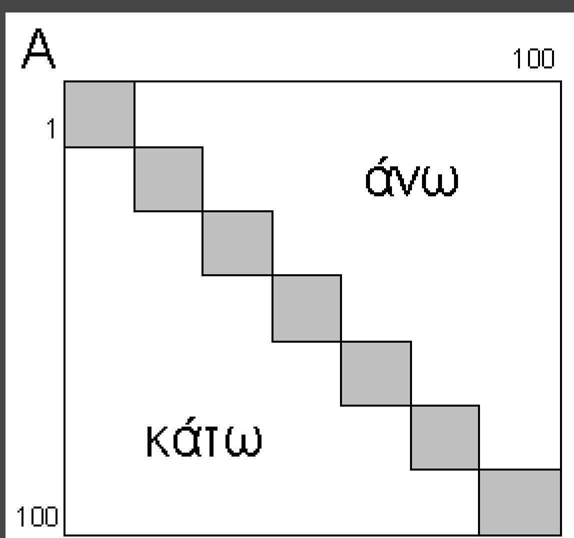


Παράδειγμα: αθροίσματα 1ης και 2ης διαγωνίου:




```
S1 ← 0
S2 ← 0
για i από 1 μέχρι 100
  S1 ← S1 + A[i,i]
  S2 ← S2 + A[i,101-i]
ΤέλοςΕπανάληψης
Γράψε S1, S2
```

Αθροίσματα άνω και κάτω της 1ης κύριας διαγωνίου:



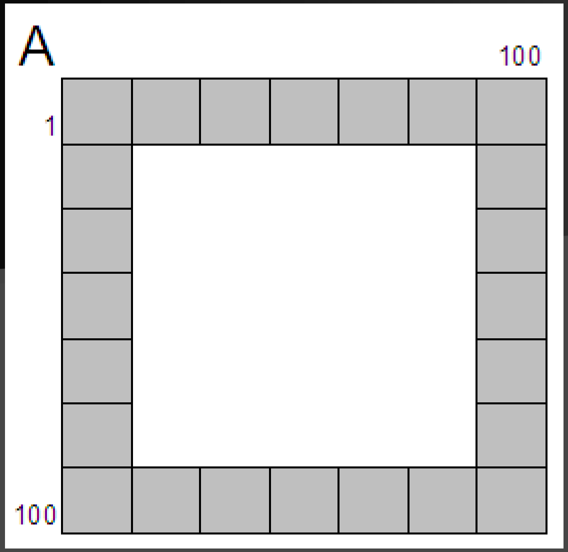

```
S1 ← 0
S2 ← 0
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i < j τότε
      S1 ← S1 + A[i,j]
    ΑλλιώςΑν i > j τότε
      S2 ← S2 + A[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε S1, S2
```

Να γεμίσετε τον A[5,5] με τις παρακάτω τιμές:

A	1	2	3	4	5
1	1	1	2	3	4
2	1	2	5	6	7
3	2	3	3	8	9
4	4	5	6	4	10
5	7	8	9	10	5

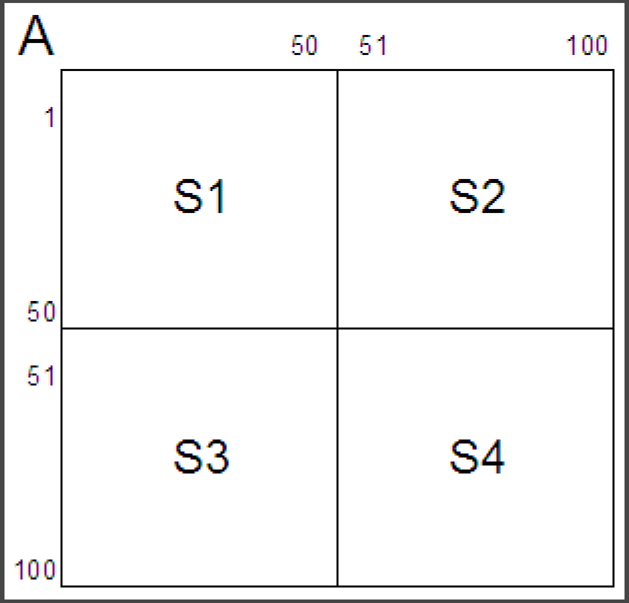
```
anw ← 0
katw ← 0
για i από 1 μέχρι 5
  για j από 1 μέχρι 5
    Αν i=j τότε
      A[i, j] ← i
    ΑλλιώςΑν i < j τότε
      anw ← anw + 1
      A[i, j] ← anw
    Αλλιώς
      katw ← katw + 1
      A[i, j] ← katw
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Άθροισμα της περιφέρειας:



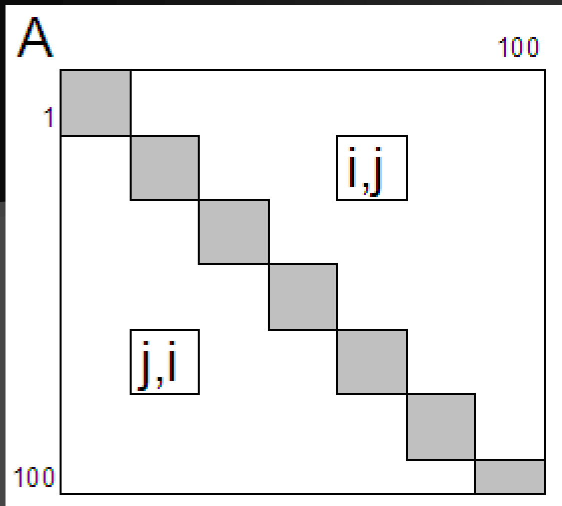

```
S ← 0
για i από 1 μέχρι 100
      S ← S + A[1,i] + A[100,i] + A[i,1] + A[i,100]
ΤέλοςΕπανάληψης
Γράψε S – (A[1,1] + A[1,100] + A[100,1] + A[100,100])
```

Αθροίσματα των τεταρτημορίων:



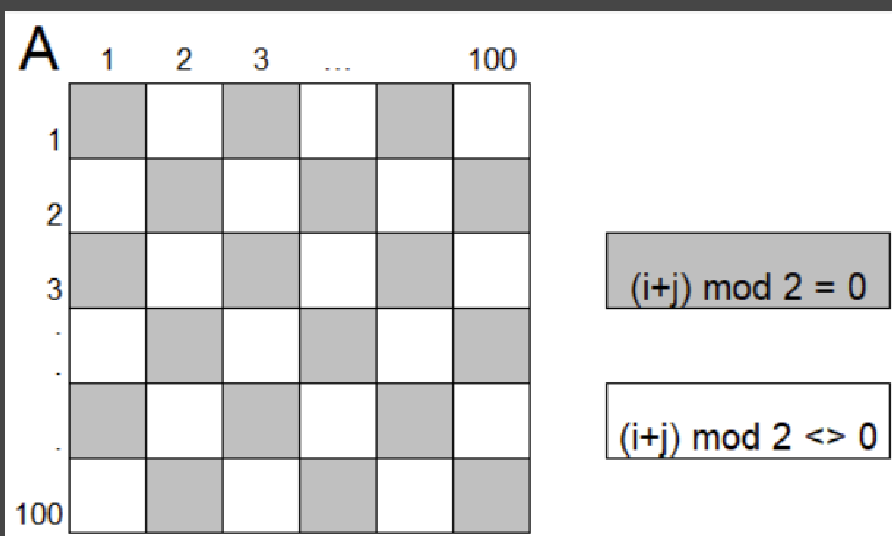

```
S1 ← 0
S2 ← 0
S3 ← 0
S4 ← 0
για i από 1 μέχρι 50
  για j από 1 μέχρι 50
      S1 ← S1 + A[i, j]
      S2 ← S2 + A[i, 50+j]
      S3 ← S3 + A[50+i, j]
      S4 ← S4 + A[50+i, 50+j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε S1, S2, S3, S4
```

Πλήθος ζευγών συμμετρικών κελιών ως προς την 1η διαγώνιο που είναι ίσα:




```
Π ← 0
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν i < j ΚΑΙ A[i,j] = A[j,i] τότε
      Π ← Π + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε Π
```

Αθροίσματα των «γκρι» και των «λευκών» κελιών:




```
S1 ← 0
S2 ← 0
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Αν (i + j) mod 2 = 0 τότε
      S1 ← S1 + A[i,j]
    Αλλιώς
      S2 ← S2 + A[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε S1, S2
```

Άλλα παραδείγματα ασκήσεων με Τετραγωνικούς Πίνακες

- Πρωτάθλημα Ν ομάδων → Α[Ν,Ν] με τα αποτελέσματα των μεταξύ τους 2 αγώνων (N/I/H)
- Χιλιομετρικές αποστάσεις Ν πόλεων → Χ[Ν,Ν]
- Διαγωνισμός Ν χωρών με βαθμολογίες → Β[Ν,Ν]
- Συμβατότητα ομάδων αίματος → Σ[8,8] (1/0)
- Διασυνδέσεις κοινωνικού δικτύου
- Ανταποκρίσεις αεροδρομίων
- Ισοτιμίες νομισμάτων

Γέμισμα του Α[100,100] με τις τιμές:

A	1		...		50	51		...		100
1	1									2
		1							2	
			1					2		
				1			2			
50					1	2				
51					3	4				
				3			4			
			3					4		
		3							4	
100	3									4

για i από 1 μέχρι …
  Α[……………………] ← 1
  Α[……………………] ← 2
  Α[……………………] ← 3
  Α[……………………] ← 4
ΤέλοςΕπανάληψης

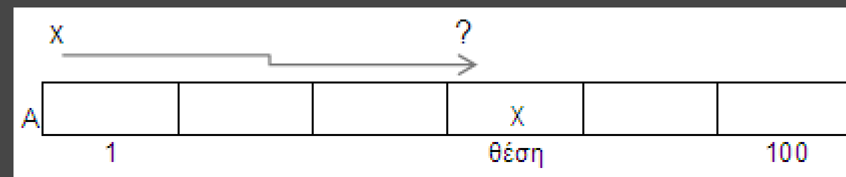
για i από 1 μέχρι 50
  A[i, i] ← 1
  A[i, 101-i] ← 2
  A[i+50, 51-i] ← 3
  A[i+50, i+50] ← 4
ΤέλοςΕπανάληψης

Έστω ο Α[100, 100]. Υπάρχει γραμμή i που είναι ίδια (έχει τις ίδιες τιμές) με την αντίστοιχη στήλη i;

```
υπάρχει <-- Ψευδής
για i από 1 μέχρι 100
  π <-- 0
  για j από 1 μέχρι 100
    Αν A[i,j] = A[j,i] τότε
      π <-- π + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν π = 100 τότε
    υπάρχει <-- Αληθής
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε υπάρχει
```

Αναζήτηση τιμής σε πίνακα

(§3.6) Σειριακή αναζήτηση – εντοπισμός της 1ης εμφάνισης (ανήκει η τιμή στον πίνακα;)




```
Διάβασε x
βρ ← Ψευδής
i ← 1
Όσο i <= 100 ΚΑΙ βρ = Ψευδής επανάλαβε
  Αν x = A[i] τότε
    βρ ← Αληθής
    θέση ← i
  Αλλιώς
    i ← i + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν βρ = Αληθής τότε
  Γράψε 'Βρέθηκε στο κελί: ', θέση
Αλλιώς
  Γράψε 'Δεν βρέθηκε'
ΤέλοςΑν
```

Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερη αποτελεσματική μέθοδος αναζήτησης. Έτσι, δικαιολογείται η χρήση της μόνο σε περιπτώσεις όπου:

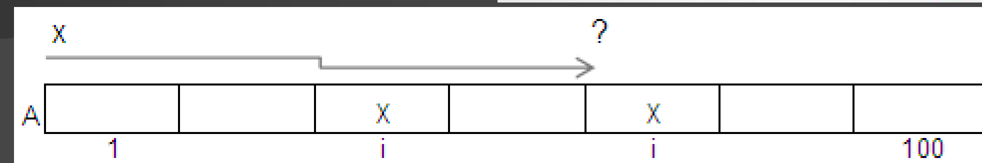
- ✓ ο πίνακας είναι μη ταξινομημένος,
- ✓ ο πίνακας είναι μικρού μεγέθους (για παράδειγμα, $n \le 20$),
- ✓ η αναζήτηση σε ένα συγκεκριμένο πίνακα γίνεται σπάνια,

Η επιλογή μεθόδου αναζήτησης σε πίνακα που εξαρτάται από το:

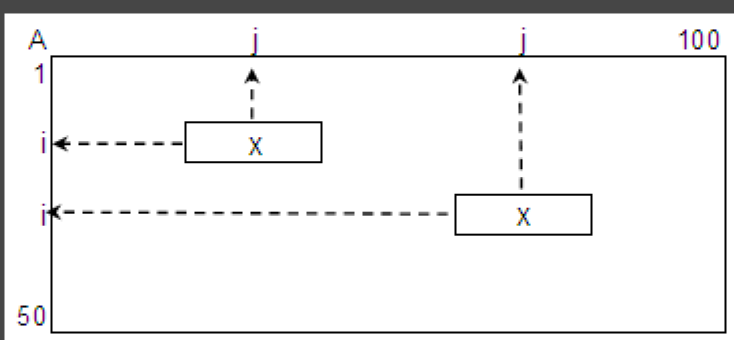
- αν ο πίνακας είναι ταξινομημένος
- αν ο πίνακας περιέχει στοιχεία που είναι όλα διαφορετικά μεταξύ τους

Εντοπισμός όλων των εμφανίσεων

```
Διάβασε x
βρ ← Ψευδής
για i από 1 μέχρι 100
  Αν x = A[i] τότε
    βρ ← Αληθής
    Γράψε ‘Βρέθηκε στο κελί: ‘, i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν βρ = Ψευδής τότε
  Γράψε ‘Δεν βρέθηκε’
ΤέλοςΑν
```



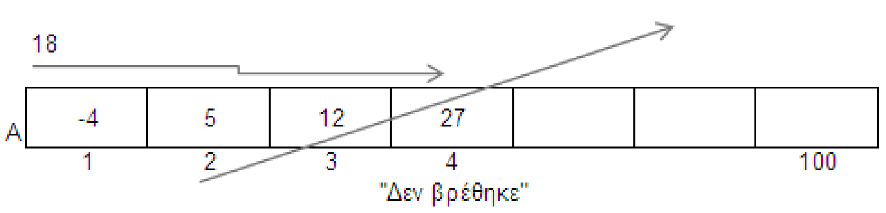
Εντοπισμός όλων των εμφανίσεων σε 2-Δ πίνακα




```
Διάβασε x
βρ ← Ψευδής
για i από 1 μέχρι 50
  για j από 1 μέχρι 100
    Αν x = A[i, j] τότε
      βρ ← Αληθής
      Γράψε ‘Βρέθηκε στη γραμμή: ‘, i, ‘ στήλη: ‘, j
    ΤέλοςΑν
ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Αν βρ = Ψευδής τότε
  Γράψε “Δεν βρέθηκε”
ΤέλοςΑν
```

Αναζήτηση τιμής σε πίνακα

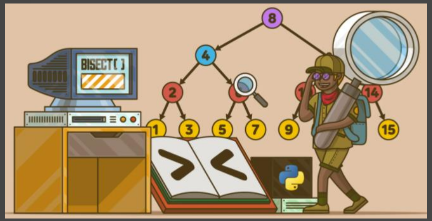
Βελτιωμένη σειριακή αναζήτηση – σε ταξινομημένο πίνακα




```
Διάβασε x
βρ <-- Ψευδής
διακοπή <-- Ψευδής
i <-- 1
Όσο i <= 100 ΚΑΙ βρ = Ψευδής ΚΑΙ διακοπή = Ψευδής επανάλαβε
  Αν x = A[i] τότε
    βρ <-- Αληθής
    θέση <-- i
  ΑλλιώςΑν x < A[i] τότε
    διακοπή <-- Αληθής
  Αλλιώς
    i <-- i + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν βρ = Αληθής τότε
  Γράψε 'Βρέθηκε στο κελί: ', θέση
Αλλιώς
  Γράψε 'Δεν βρέθηκε'
ΤέλοςΑν
```

Εισαγωγή τιμής με εγκυρότητα ώστε να ανήκει σε πίνακα

```
ΑρχήΕπανάληψης
  Διάβασε x
  βρ <-- Ψευδής
  i <-- 1
  Όσο i <= 100 ΚΑΙ βρ = Ψευδής επανάλαβε
    Αν x = A[i] τότε
      βρ <-- Αληθής
    Αλλιώς
      i <-- i + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΜέχριςΌτου βρ = Αληθής
```

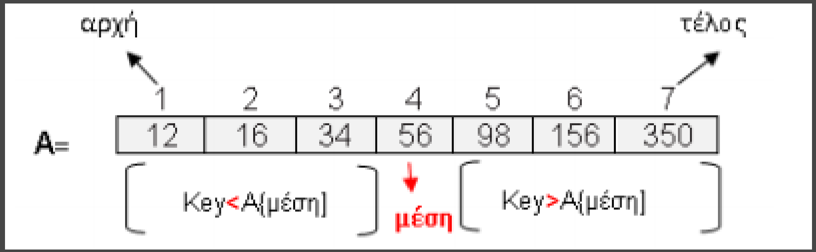


Αναζήτηση τιμής σε πίνακα

Δυαδική αναζήτηση – ο πίνακας πρέπει να είναι ταξινομημένος π.χ. σε αύξουσα σειρά

Στοιχεία Ν	Συγκρίσεις
10	4
100	7
1,000	10
10,000	14
100,000	17
1,000,000	20
10,000,000	24
100,000,000	27
1,000,000,000	30

```
ΔΙΑΒΑΣΕ Key
 arxh ←  1
 telos ←  N
 mesh  ← (arxh + telos) div 2
 found ← ΨΕΥΔΗΣ
 ΟΣΟ arxh <= telos ΚΑΙ found = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
   ΑΝ Key < A[mesh] ΤΟΤΕ
     telos ← mesh-1 ! για ↓ ταξινόμηση: Αρχή ← μέση + 1
   ΑΛΛΙΩΣ_ΑΝ Key > A[mesh] ΤΟΤΕ
     arxh ← mesh + 1 ! για ↓ ταξινόμηση: Τέλος ← μέση-1
   ΑΛΛΙΩΣ
     found ← ΑΛΗΘΗΣ
   ΤΕΛΟΣ_ΑΝ
   mesh ←  (arxh + telos) div 2
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΑΝ found = ΑΛΗΘΗΣ ΤΟΤΕ
   ΓΡΑΨΕ 'Βρέθηκε στη θέση ', mesh
 ΑΛΛΙΩΣ
   ΓΡΑΨΕ 'Δεν βρέθηκε'
 ΤΕΛΟΣ_ΑΝ
```

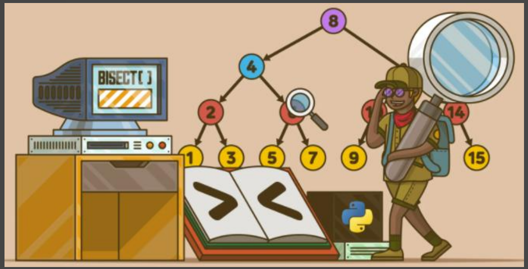


Εάν ο πίνακας περιέχει διπλά στοιχεία, ο αλγόριθμος δυαδικής αναζήτησης ενδέχεται να μην επιστρέψει το δείκτη της πρώτης εμφάνισης του στοιχείου. Μπορεί να επιστρέψει το δείκτη οποιουδήποτε στοιχείου (και όχι όλων).

Αναζήτηση τιμής σε πίνακα

Δυαδική αναζήτηση – Παράδειγμα: Δίνεται ο πίνακας

1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



Σε ακέραιο A[100] γεμάτο με τιμές 1-100, αναζήτηση της τιμής:
78
1 A[mesh]= 50 => pio panw
2 A[mesh]= 75 => pio panw
3 A[mesh]= 88 => pio katw
4 A[mesh]= 81 => pio katw
5 A[mesh]= 78 => bingo
100
1 A[mesh]= 50 => pio panw
2 A[mesh]= 75 => pio panw
3 A[mesh]= 88 => pio panw
4 A[mesh]= 94 => pio panw
5 A[mesh]= 97 => pio panw
6 A[mesh]= 99 => pio panw
7 A[mesh]= 100 => bingo
1
1 A[mesh]= 50 => pio katw
2 A[mesh]= 25 => pio katw
3 A[mesh]= 12 => pio katw
4 A[mesh]= 6 => pio katw
5 A[mesh]= 3 => pio katw
6 A[mesh]= 1 => bingo

Αναζήτηση του στοιχείου 38 (υπάρχει στον πίνακα)

Βήμα 1	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 2	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 3	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 4	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47

Με κίτρινο σημειώνεται το στοιχείο του πίνακα που εξετάζεται (στο μέσον)

Με πράσινο σημειώνεται το τμήμα του πίνακα που απομένει για αναζήτηση

Με κόκκινο σημειώνεται το τμήμα του πίνακα που έχει αποκλειστεί

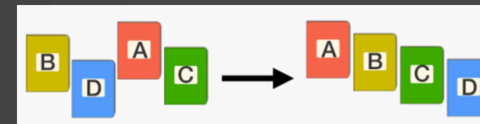
Αναζήτηση του στοιχείου 39 (δεν υπάρχει στον πίνακα)

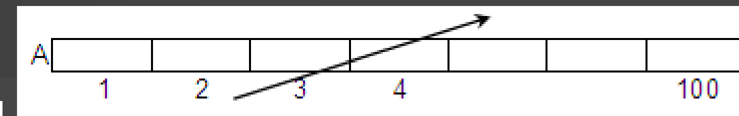
Βήμα 1	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 2	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 3	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 4	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47
Βήμα 5	1	2	5	8	9	15	22	27	35	37	38	40	43	45	47

Ταξινόμηση ευθείας ανταλλαγής (ή φυσαλίδας ή bubble sort)

Ορισμός. Δοθέντων των στοιχείων $a_1, a_2, \ldots, a_n$ η ταξινόμηση συνίσταται στη *μετάθεση* (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά $a_{k1}, a_{k2}, \ldots, a_{kn}$ έτσι ώστε, δοθείσης μίας *συνάρτησης διάταξης* (ordering function), f, να ισχύει:

$$f(a_{k1}) \leq f(a_{k2}) \leq \ldots \leq f(a_{kn})$$





52	5	5	5	5	5	5	5	5
12	52	10	10	10	10	10	10	10
71	12	52	12	12	12	12	12	12
56	71	12	52	19	19	19	19	19
5	56	71	19	52	45	45	45	45
10	10	56	71	45	52	52	52	52
19	19	19	56	71	56	56	56	56
90	45	45	45	56	71	71	71	71
45	90	90	90	90	90	90	90	90

```
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν A[j-1] > A[j] τότε
      tmp ← A[j-1]
      A[j-1] ← A[j]
      A[j] ← tmp
      ! ή Αντιμετάθεσε(A[j], A[j-1])
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Για την ταξινόμηση δεδομένων έχουν εκπονηθεί πάρα πολλοί αλγόριθμοι. Άλλοι σχετικά απλοί αλγόριθμοι είναι η ταξινόμηση με επιλογή και η ταξινόμηση με παρεμβολή. Ο πιο γρήγορος αλγόριθμος ταξινόμησης είναι η "γρήγορη ταξινόμηση" (quicksort). Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο αργός αλγόριθμος ταξινόμησης.

Επαναλήψεις:

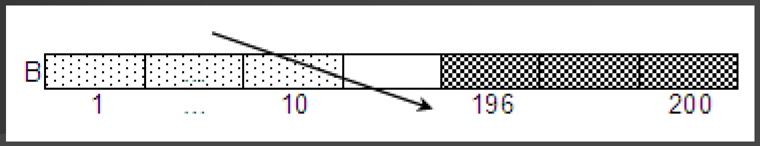
$$N - 1 + N - 2 + \cdots + 2 + 1 = \frac{N \cdot (N-1)}{2}$$

Ένα κριτήριο επιλογής αλγορίθμου ταξινόμησης είναι η αρχική διάταξη των στοιχείων του πίνακα

για φθίνουσα ταξινόμηση:
    Αν A[j-1] < A[j] τότε

Ταξινόμηση ευθείας ανταλλαγής (ή φυσσαλίδας ή bubblesort)

```
για i από 2 μέχρι 200
  για j από 200 μέχρι i μεβήμα -1
    Αν B[j-1] < B[j] τότε
      tmp ← B[j-1]
      B[j-1] ← B[j]
      B[j] ← tmp
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε ‘10 υψηλότεροι:’
για i από 1 μέχρι 10
  Γράψε B[i]
ΤέλοςΕπανάληψης
Γράψε ‘5 χαμηλότεροι:’
για i από 196 μέχρι 200
  Γράψε B[i]
ΤέλοςΕπανάληψης
```

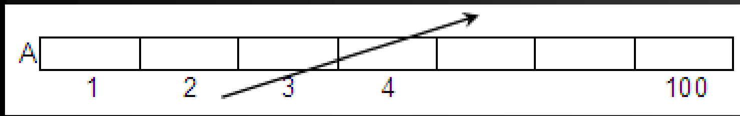


π.χ.1) δίνεται ο B[200] με τους βαθμούς των απολυτηρίων 200 μαθητών. Ποιοί είναι οι 10 υψηλότεροι και οι 5 χαμηλότεροι βαθμοί;

Διαβάζει έναν ακέραιο Ν (1-200) και υπολογίζει τη διάμεσο των Ν καλύτερων βαθμών (διάμεσος: η μεσαία τιμή, αν Ν περιττός ή το ημιάθροισμα 2 μεσαίων τιμών, αν Ν άρτιος)

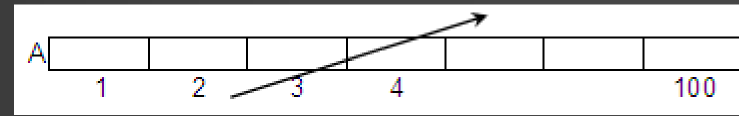

```
ΑρχήΕπανάληψης
  Διάβασε Ν
ΜέχριςΌτου Ν>=1 ΚΑΙ Ν<=200
Αν Ν mod 2 = 1 τότε
  διάμεσος <-- B[N div 2 + 1]
Αλλιώς
  διάμεσος <-- (B[N div 2] + B[N div 2 + 1])/2
ΤέλοςΑν
Γράψε διάμεσος
```

Ταξινόμηση ευθείας ανταλλαγής (ή φυσσαλίδας ή bubblesort)



π.χ. 2 Δίνεται ο ακέραιος A[100]. Ποιες και πόσες οι διαφορετικές τιμές του;

```
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν A[j-1] > A[j] τότε
      tmp <-- A[j-1]
      A[j-1] <-- A[j]
      A[j] <-- tmp
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Π <-- 1
Γράψε A[1]
για i από 2 μέχρι 100
  Αν A[i] <> A[i-1] τότε
    Γράψε A[i]
    Π <-- Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε 'Πλήθος τιμών:', Π
```

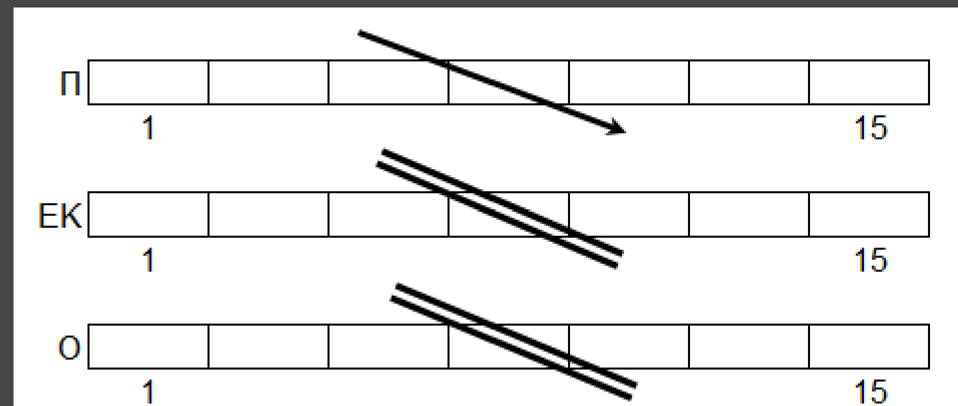


π.χ. 3 Δίνεται ο ακέραιος A[100]. Ποιες οι πιο "συχνές" τιμές του;

```
! ταξινόμηση του A[100]...
seri <-- 1
maxSeri <-- 1
για i από 2 μέχρι 100
  Αν A[i] = A[i-1] τότε
    seri <-- seri + 1
  Αλλιώς
    seri <-- 1
  ΤέλοςΑν
  Αν seri > maxSeri  τότε
    maxSeri <-- seri
  ΤέλοςΑν
ΤέλοςΕπανάληψης
seri <-- 1
για i από 2 μέχρι 100
  Αν A[i] = A[i-1] τότε
    seri <-- seri + 1
  Αλλιώς
    seri <-- 1
  ΤέλοςΑν
  Αν seri = maxSeri  τότε
    Γράψε A[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Ταξινόμηση παράλληλων 1-Δ πινάκων

α) με 1 κλειδί ταξινόμησης




```
για i από 2 μέχρι 15
  για j από 15 μέχρι i μεβήμα -1
    Αν Π[j-1] < Π[j] τότε
      tmp1 ← Π[j-1]
      Π[j-1] ← Π[j]
      Π[j] ← tmp1
      tmp2 ← ΕΚ[j-1]
      ΕΚ[j-1] ← ΕΚ[j]
      ΕΚ[j] ← tmp2
      tmp3 ← Ο[j-1]
      Ο[j-1] ← Ο[j]
      Ο[j] ← tmp3
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 15
  Γράψε Ο[i], Π[i], ΕΚ[i]
ΤέλοςΕπανάληψης
```

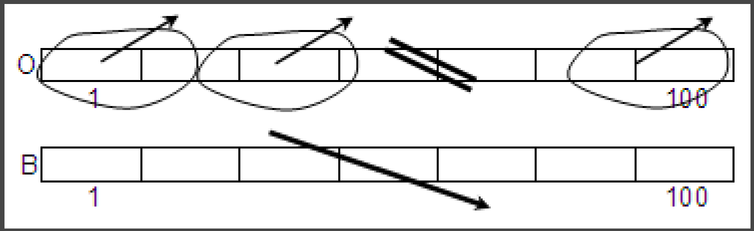
π.χ. Δίνονται οι Ο[15], Π[15] και ΕΚ[15], με τα ονόματα, τους πληθυσμούς και τις εκτάσεις 15 Ευρωπαϊκών χωρών. Να εμφανισθούν κατά φθίνουσα σειρά πληθυσμών

Ταξινόμηση παράλληλων 1-Δ πινάκων

β) με 2 κλειδιά ταξινόμησης

π.χ. Δίνονται οι Ο[100] και Β[100] με τα ονόματα και τους βαθμούς του απολυτηρίου 100 μαθητών. Να εμφανισθούν κατά φθίνουσα σειρά βαθμών απολυτηρίου και όπου υπάρχει ισοβαθμία, κατά αλφαβητική σειρά

```
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν Β[j-1] < Β[j] τότε
      tmp1 ← Β[j-1]
      Π[j-1] ← Β[j]
      Β[j] ← tmp1
      tmp2 ← Ο[j-1]
      Ο[j-1] ← Ο[j]
      Ο[j] ← tmp2
    ΑλλιώςΑν Β[j-1] = Β[j] ΚΑΙ Ο[j-1] > Ο[j] τότε
      tmp2 ← Ο[j-1]
      Ο[j-1] ← Ο[j]
      Ο[j] ← tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 100
  Γράψε Ο[i], Β[i]
ΤέλοςΕπανάληψης
```



Όνομα	Βαθμός
Ρ	15
Λ	19
Π	15
Κ	19
Γ	18

⇒

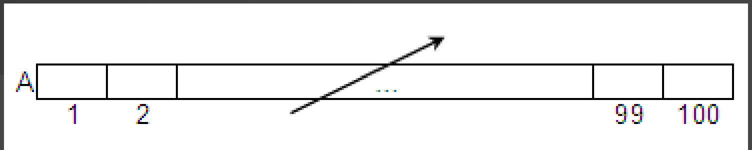
Όνομα	Βαθμός
Κ	19
Λ	19
Γ	18
Π	15
Ρ	15

Ταξινόμηση

Έλεγχος για ταξινόμηση π.χ. δίνεται ο A [100]. Να ελεγχθεί αν είναι ταξινομημένος κατά αύξουσα σειρά

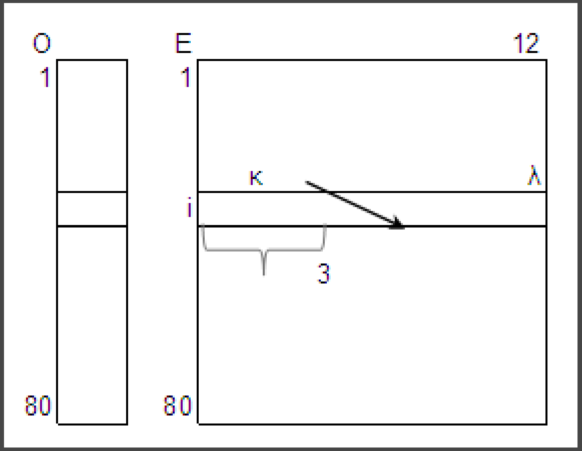
```
Π ← 0
για i από 2 μέχρι 100
  Αν A[i] >= A[i-1] τότε
    Π ← Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
Αν Π = 99 τότε
  Γράψε "ναι"
Αλλιώς
  Γράψε "όχι"
ΤέλοςΑν
```



Ταξινόμηση των γραμμών 2-Δ πίνακα π.χ. δίνεται ο Ε[80, 12] με τις μηνιαίες εισπράξεις 80 επιχειρήσεων για 1 έτος και ο Ο[80] με τα ονόματά τους. Να εμφανισθούν οι επιχειρήσεις ακολουθούμενες από τις 3 μεγαλύτερες μηνιαίες εισπράξεις της κάθε μίας

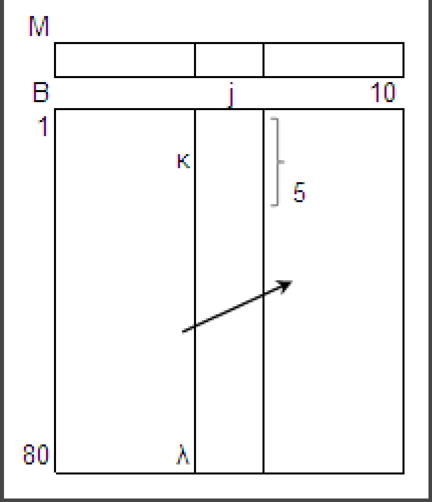
```
για i από 1 μέχρι 80
  για κ από 2 μέχρι 12
    για λ από 12 μέχρι κ μεβήμα -1
      Αν Ε[i, λ-1] < Ε[i, λ] τότε
        tmp ← Ε[i, λ-1]
        Ε[i, λ-1] ← Ε[i, λ]
        Ε[i, λ] ← tmp
      ΤέλοςΑν
    ΤέλοςΕπανάληψης
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 80
  Γράψε Ο[i]
  για j από 1 μέχρι 3
    Γράψε Ε[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```



Ταξινόμηση

Ταξινόμηση των στηλών 2-Δ πίνακα π.χ. δίνεται ο B[80, 10] με τους βαθμούς 80 μαθητών σε 10 μαθήματα και ο M[10] με τα ονόματα των μαθημάτων. Να εμφανισθούν τα μαθήματα ακολουθούμενα από τους 5 χαμηλότερους βαθμούς του καθενός

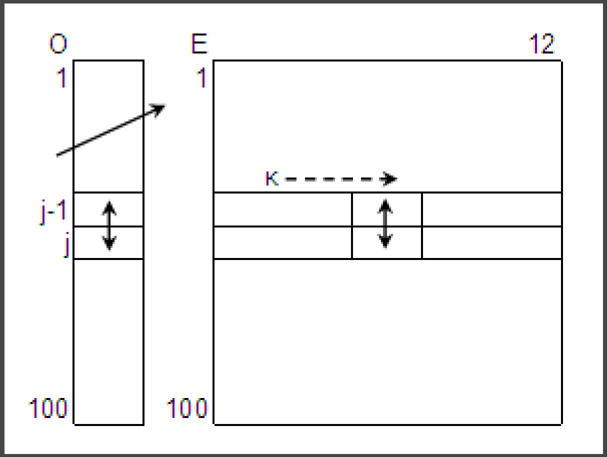
```
για j από 1 μέχρι 10
  για κ από 2 μέχρι 80
    για λ από 80 μέχρι κ μεβήμα -1
      Αν B[λ-1, j] > B[λ, j] τότε
        tmp ← B[λ-1, j]
        B[λ-1, j]  ← B[λ, j]
        B[λ, j]  ← tmp
      ΤέλοςΑν
    ΤέλοςΕπανάληψης
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για j από 1 μέχρι 10
  Γράψε M[j]
  για i από 1 μέχρι 5
    Γράψε B[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```



Ταξινόμηση

Παράλληλη ταξινόμηση 1-Δ και 2-Δ πίνακα κατά γραμμές π.χ. δίνεται ο Ο[100] με τα ονόματα εταιρειών και ο Ε[100, 12] με τις μηνιαίες εισπράξεις τους για 1 έτος. Να εμφανισθούν κατά αλφαβητική σειρά εταιρειών (↑ Ο[100])

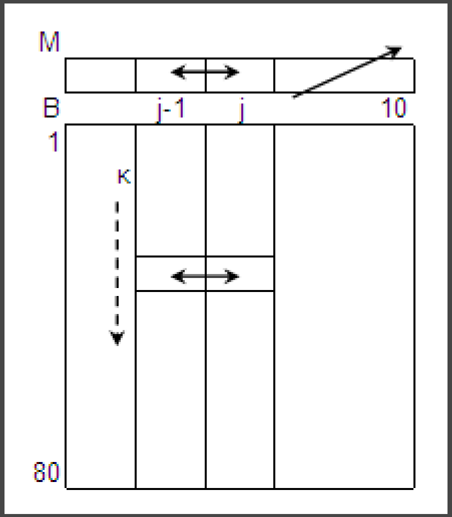
```
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν Ο[j-1] > Ο[j] τότε
      tmp ← Ο[j-1]
      Ο[j-1] ← Ο[j]
      Ο[j] ← tmp
      ! αντιμετάθεση των γραμμών j, j-1 του Ε[100, 12]
      για κ από 1 μέχρι 12
        tmp2 ← Ε[j-1, κ]
        Ε[j-1, κ] ← Ε[j, κ]
        Ε[j, κ] ← tmp2
      ΤέλοςΕπανάληψης
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 100
  Γράψε Ο[i]
  για j από 1 μέχρι 12
    Γράψε Ε[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```



Ταξινόμηση

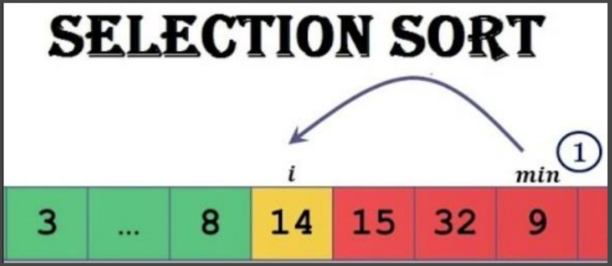
Παράλληλη ταξινόμηση 1-Δ και 2-Δ πίνακα κατά στήλες π.χ. δίνεται ο Β[80, 10] με τους βαθμούς 80 μαθητών σε 10 μαθήματα και ο Μ[10] με τα ονόματα των μαθημάτων. Να εμφανισθούν κατά αλφαβητική σειρά μαθημάτων (↑ Μ[10])

```
για i από 2 μέχρι 10
  για j από 10 μέχρι i μεβήμα -1
    Αν Μ[j-1] > M[j] τότε
      tmp ← M[j-1]
      M[j-1] ← M[j]
      M[j] ← tmp
      ! αντιμετάθεση των στηλών j, j-1 του Β[80, 10]
      για κ από 1 μέχρι 80
        tmp2 ← B[κ, j-1]
        B[κ, j-1] ← B[κ, j]
        B[κ, j] ← tmp2
      ΤέλοςΕπανάληψης
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για j από 1 μέχρι 10
  Γράψε Μ[j]
  για i από 1 μέχρι 80
    Γράψε Β[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

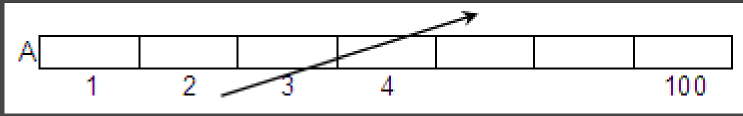


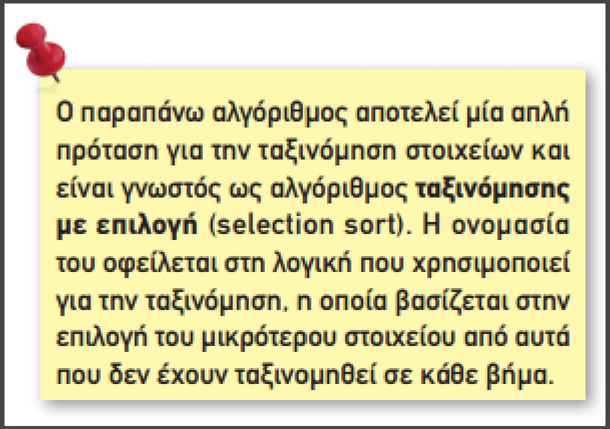
Ταξινόμηση 2Δ πίνακα.
Δίνεται ο ακέραιος Α[50, 80]. Να εμφανισθούν οι τιμές του κατά αύξουσα σειρά.
Λύση: α) αντιγραφή του Α[50, 80] στον Β[4000]
β) ταξινόμηση και εμφάνιση του Β.

Ταξινόμηση με επιλογή – Selection sort - *(Τετράδιο Μαθητή)*




```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 99
  ! βρες το μικρότερο στοιχείο και τη θέση του ελαχίστου
  ! από τη θέση i μέχρι το τέλος του πίνακα
  min <- A[i]
  θmin <- i
  ΓΙΑ j ΑΠΟ i + 1 ΜΕΧΡΙ 100
    ΑΝ A[j] < min ΤΟΤΕ
      min <- A[j]
      θmin <- j
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! αντιμετάθεσε στοιχεία θέσεων i και θmin
  temp <- A[i]
  A[i] <- A[θmin]
  A[θmin] <- temp
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```




Ο παραπάνω αλγόριθμος αποτελεί μία απλή πρόταση για την ταξινόμηση στοιχείων και είναι γνωστός ως αλγόριθμος **ταξινόμησης με επιλογή** (selection sort). Η ονομασία του οφείλεται στη λογική που χρησιμοποιεί για την ταξινόμηση, η οποία βασίζεται στην επιλογή του μικρότερου στοιχείου από αυτά που δεν έχουν ταξινομηθεί σε κάθε βήμα.

! για φθίνουσα ταξινόμηση: βρες το μέγιστο στοιχείο και τη θέση του μεγίστου

Ταξινόμηση με επιλογή - Selection sort - *(Παράδειγμα)*

Αν υποθέσουμε ότι έχουμε το πίνακα A[8]:

46	55	12	42	94	18	06	67

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

46	55	12	42	94	18	06	67

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

06	55	12	42	94	18	46	67

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

06	12	55	42	94	18	46	67

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

06	12	18	42	94	55	46	67

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

06	12	18	42	94	55	46	67

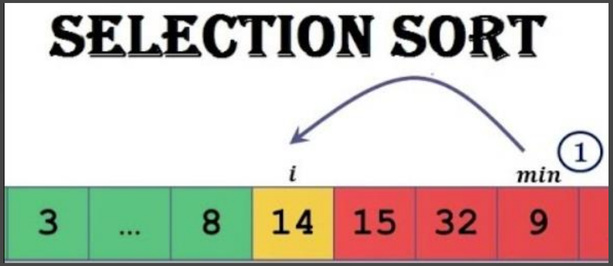
Βήμα 6 (επανάληψη της ανωτέρω διαδικασίας αλλά στο τμήμα του πίνακα από το έκτο στοιχείο και κάτω)

06	12	18	42	46	55	94	67

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

06	12	18	42	46	55	94	67

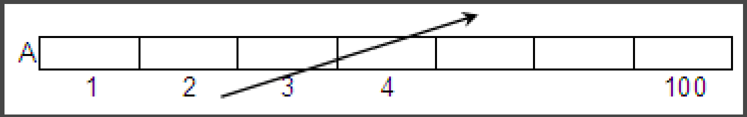
06	12	18	42	46	55	67	94

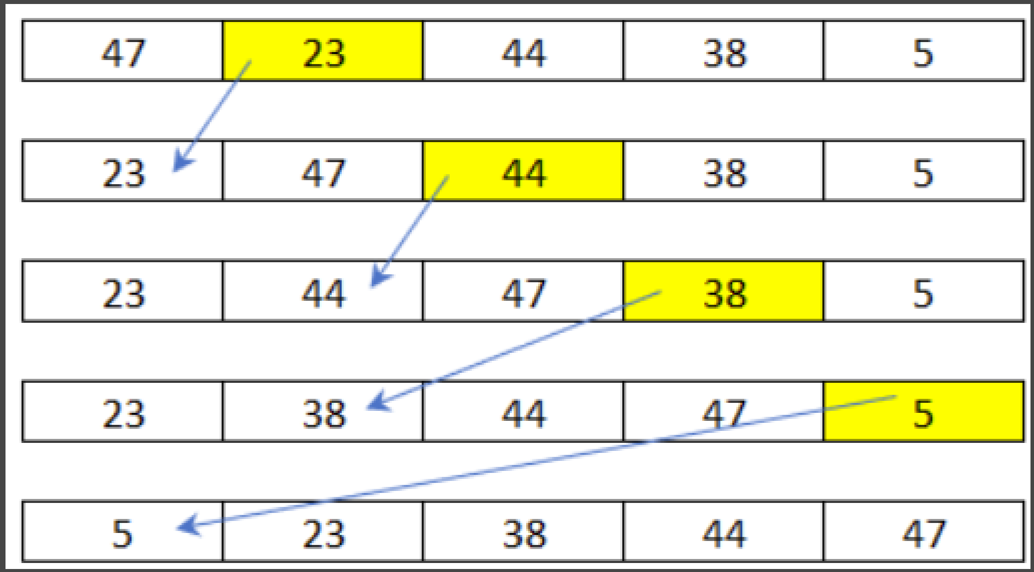


Ταξινόμηση με εισαγωγή– Insertion sort

Σε κάθε επανάληψη τοποθετείται κάθε στοιχείο του πίνακα στη σωστή θέση σε σχέση με τα προηγούμενα, ώστε το τμήμα του πίνακα μέχρι το στοιχείο αυτό να παραμένει ταξινομημένο.

```
για i από 2 μέχρι 100
  j ← i
  Όσο j > 1 ΚΑΙ A[j-1] > A[j] επανάλαβε
    tmp ← A[j]
    A[j] ← A[j-1]
    A[j-1] ← A[j]
    j ← j - 1
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

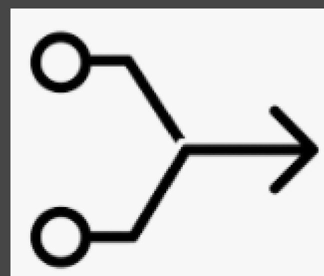


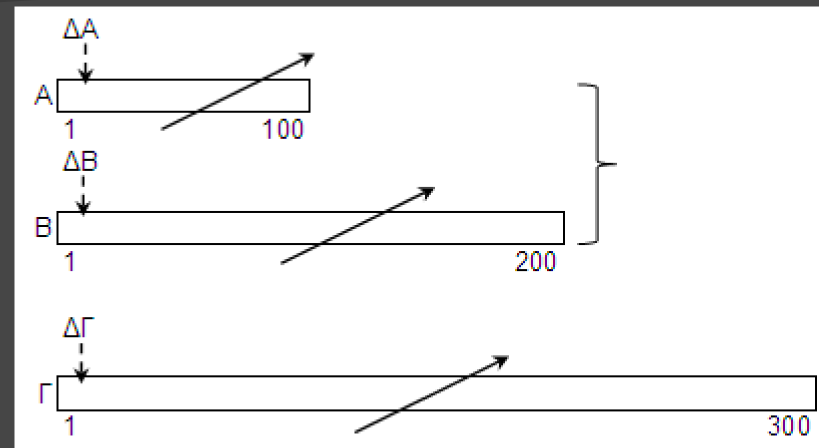


! για φθίνουσα ταξινόμηση: Όσο j > 1 ΚΑΙ A[j-1] < A[j]

Συγχώνευση δύο ταξινομημένων 1-Δ πινάκων σε 3ο (ταξινομημένος)

```
ΔΑ ← 1 ΔΒ ← 1 ΔΓ ← 1
Όσο ΔΑ <= 100 ΚΑΙ ΔΒ <= 200 επανάλαβε
  Αν Α[ΔΑ] < Β[ΔΒ] τότε
    Γ[ΔΓ] ← Α[ΔΑ]
    ΔΑ ← ΔΑ + 1
  Αλλιώς
    Γ[ΔΓ] ← Β[ΔΒ]
    ΔΒ ← ΔΒ + 1
  ΤέλοςΑν
  ΔΓ ← ΔΓ + 1
ΤέλοςΕπανάληψης
Αν ΔΑ <= 100 τότε ! περίσσευμα στον Α
  για i από ΔΑ μέχρι 100
    Γ[ΔΓ] ← Α[i]
    ΔΓ ← ΔΓ + 1
  ΤέλοςΕπανάληψης
Αλλιώς ! περίσσευμα στον Β
  για i από ΔΒ μέχρι 200
    Γ[ΔΓ] ← Β[i]
    ΔΓ ← ΔΓ + 1
  ΤέλοςΕπανάληψης
ΤέλοςΑν
```



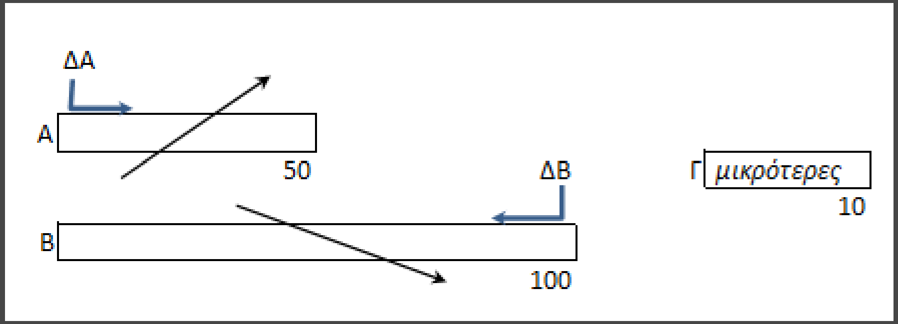


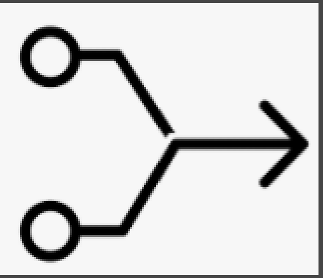
π.χ. Δίνονται οι Α[100] και Β[200] ταξινομημένοι κατά αύξουσα σειρά. Να συγχωνευθούν στον Γ[300] ώστε να είναι και αυτός ταξινομημένος.

Συλλογή των Ν μεγαλύτερων/μικρότερων τιμών από 2 ταξινομημένους 1-Δ πίνακες

π.χ. Δίνονται οι ακέραιοι πίνακες Α[50] και Β[100] ταξινομημένοι κατά αύξουσα και φθίνουσα σειρά αντίστοιχα. Να γεμίσετε τον ακέραιο Γ[10] με τις 10 μικρότερες τιμές των Α και Β. Παρατήρηση: οι τιμές των Α και Β είναι όλες διαφορετικές μεταξύ τους.

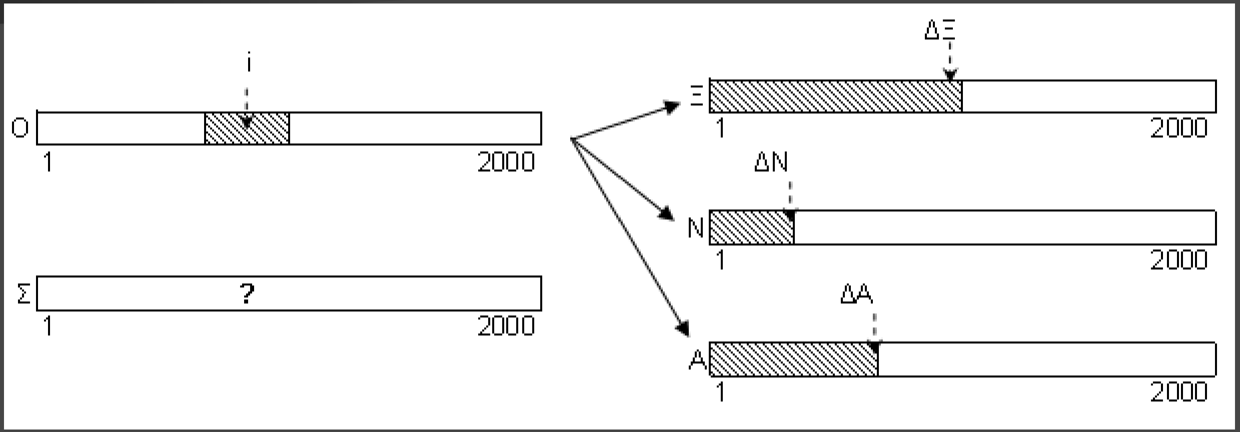
```
ΔΑ ← 1 ΔΒ ← 100
για i από 1 μέχρι 10
  Αν Α[ΔΑ] < Β[ΔΒ] τότε
    Γ[i] ← Α[ΔΑ]
    ΔΑ ← ΔΑ + 1
  Αλλιώς
    Γ[i] ← Β[ΔΒ]
    ΔΒ ← ΔΒ - 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```





Διαχωρισμός 1-Δ πίνακα

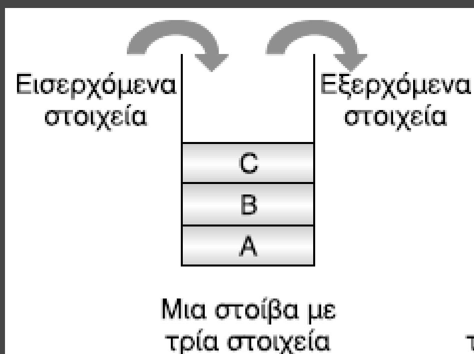
```
ΔΞ ← 0
ΔΝ ← 0
ΔΑ ← 0
για i από 1 μέχρι 2000
  Αν Σ[i] = 'Ξ' τότε
    ΔΞ ← ΔΞ + 1
    Ξ[ΔΞ] ← Ο[i]
  ΑλλιώςΑν Σ[i] = 'Ν' τότε
    ΔΝ ← ΔΝ + 1
    Ν[ΔΝ] ← Ο[i]
  Αλλιώς
    ΔΑ ← ΔΑ + 1
    Α[ΔΑ] ← Ο[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```



π.χ. Δίνονται τα ονόματα και τα σώματα ('Ξ', 'Ν', 'Α') 2000 νεοσύλλεκτων στους Ο[2000] και Σ[2000] αντίστοιχα. Να διαχωρισθούν τα ονόματα σε 3 ξεχωριστούς πίνακες ανά σώμα

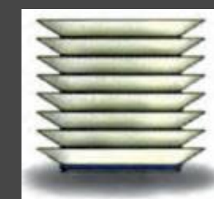
(§ 3.4) Στοίβα

Ορισμός: δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία.



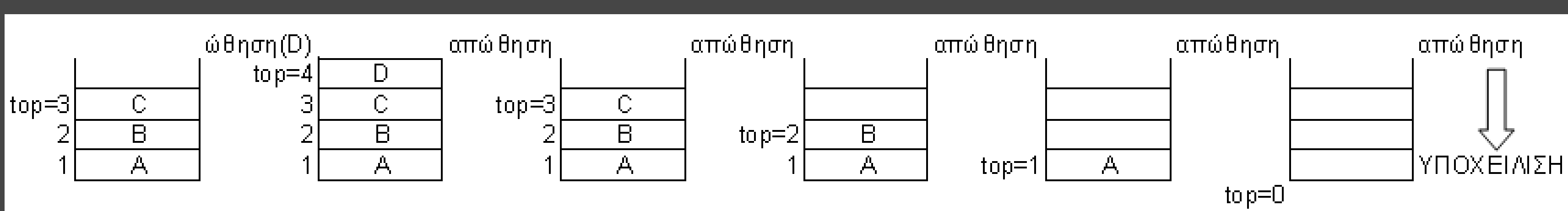
άδεια στοίβα ⇔ top = 0,
γεμάτη στοίβα ⇔ top = N

Μέθοδος επεξεργασίας: «τελευταίο μέσα, πρώτο έξω» (LIFO: last in – first out). Όπως για παράδειγμα: μία στοίβα από πιάτα, ή η στοίβα χρόνου εκτέλεσης των υποπρογραμμάτων (κεφ.10).



Κύριες λειτουργίες:

- **Ώθηση** (push): ο top αυξάνεται κατά 1 και το νέο στοιχείο ωθείται στο Σ[top]. Έλεγχος για υπερχείλιση (stack overflow) ⇔ ήθηση σε γεμάτη στοίβα
- **Απώθηση** (pop): απωθείται το κορυφαίο στοιχείο (Σ[top]) και ο top μειώνεται κατά 1. Έλεγχος για υποχείλιση (stack underflow) ⇔ απώθηση σε άδεια στοίβα



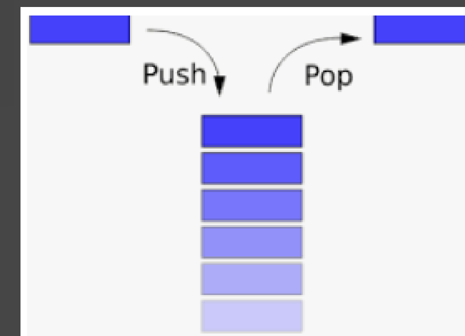
(§ 3.4) Στοίβα (έστω στοίβα Σ[1000])

Ώθηση στοιχείου x

Αν top < 1000 τότε
  top ← top + 1
  Σ[top] ← x
Αλλιώς
  Γράψε ‘υπερχείλιση’
ΤέλοςΑν

Απώθηση στοιχείου

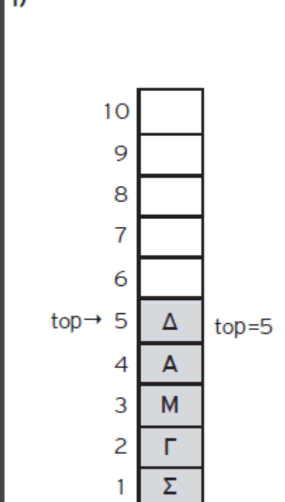
Αν top <> 0 τότε
  Γράψε Σ[top]
  top ← top-1
Αλλιώς
  Γράψε ‘υποχείλιση’
ΤέλοςΑν



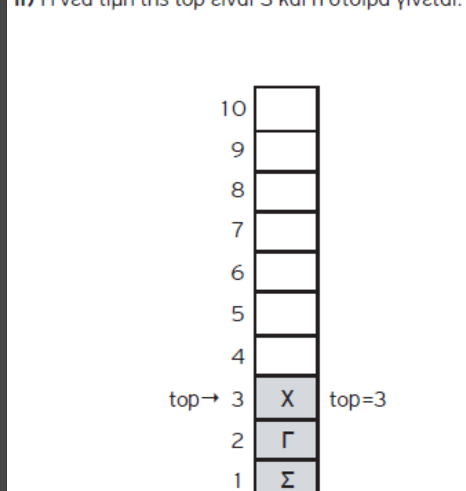
πλήθος στοιχείων στη στοίβα = top

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;

i)



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



	Στοίβα 10 θέσεων		
	ΠΡΙΝ top	λειτουργία	ΜΕΤΑ top
1	0	ώθηση	1
2	0	απώθηση	0
3	1	ώθηση	2
4	1	απώθηση	0
5	5	ώθηση	6
6	5	απώθηση	4
7	10	ώθηση	10
8	10	απώθηση	9

Ε. 1: Δίνεται η επόμενη ακολουθία αριθμών: 4, 8, 2, 5, 9, 13.

α) Ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών σε στοίβα;

β) Σχεδιάστε τη στοίβα μετά την τοποθέτηση των αριθμών.

γ) Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο των αριθμών από τη στοίβα;

δ) Πόσες φορές θα πρέπει να εκτελεστεί η προηγούμενη λειτουργία στη στοίβα για να εξαχθεί ο αριθμός 5;

Ε. 2: Σε μια στοίβα έχουν τοποθετηθεί κατά σειρά οι αριθμοί : 24, 7, 11, 13, 65, 39, 5.

α) Να σχεδιάσετε την παραπάνω δομή.

β) Ποια θα είναι η τιμή του δείκτη της παραπάνω στοίβας;

γ) Αν θέλετε να τοποθετήσετε τον αριθμό 25 στην στοίβα, ποια λειτουργία θα χρησιμοποιήσετε;

δ) Ποια θα είναι η τιμή του δείκτη μετά την λειτουργία αυτή;

ε) Αν θέλετε να εξάγετε τον αριθμό 65 από τη στοίβα, ποια λειτουργία θα χρησιμοποιήσετε;

στ) Ποια θα είναι η τιμή του δείκτη μετά τη λειτουργία αυτή;

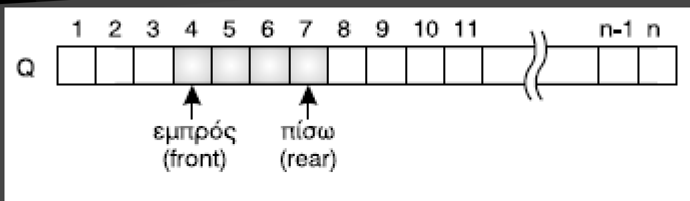
		Σ	Λ
1	Για την υλοποίηση μιας στοίβας μπορεί να χρησιμοποιηθεί ένας πίνακας.	❐	❐
2	Στη στοίβα το στοιχείο που μπαίνει πρώτο βγαίνει πρώτο.	❐	❐
3	Στην υλοποίηση της στοίβας χρειάζονται δύο μεταβλητές-δείκτες για την υλοποίηση των δύο βασικών λειτουργιών που εκτελούνται σε αυτή.	❐	❐
4	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια άδεια στοίβα.	❐	❐
5	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια γεμάτη στοίβα.	❐	❐
6	Η ώθηση στοιχείου γίνεται στην κορυφή της στοίβας.	❐	❐
7	Στη δομή της στοίβας απαιτούνται δύο δείκτες, ο εμπρός και ο πίσω.	❐	❐
8	Υπερχείλιση έχουμε όταν εισάγουμε ένα στοιχείο σε μια ήδη γεμάτη στοίβα.	❐	❐
9	Η μέθοδος LIFO περιγράφει τη διαδικασία εκείνη κατά την οποία το στοιχείο που τοποθετείται τελευταίο εξάγεται πρώτο	❐	❐
10	Κάθε στοιχείο που εισάγεται πρώτο σε μια στοίβα είναι αυτό που εξάγεται πρώτο.	❐	❐

Άσκηση στοίβας: Να γραφεί πρόγραμμα διαχείρισης ακέραιας στοίβας Σ[100]. Αρχικά να την αδειάζει. Με επαναληπτικό μενού να εκτελεί τα παρακάτω: 1. Ώθηση 2. Απώθηση 3. Πλήθος στοιχείων 4. Έξοδος. Τελικά να εμφανίζει: α) πόσες φορές γέμισε και πόσες φορές άδειασε β) πόσες φορές υπερχείλισε και πόσες φορές υποχείλισε γ) ποιο το μέγιστο πλήθος στοιχείων που είχε.

```
Πρόγραμμα Στοίβα
 Μεταβλητές
  Ακέραιες: Σ[100], top, επ, Π_γέμισε,
Π_άδειασε, Π_υπερχείλισε, Π_υποχείλισε,
max_στοιχεία
Αρχή
 top <-- 0
 Π_γέμισε <-- 0
 Π_άδειασε <-- 0
 Π_υπερχείλισε <-- 0
 Π_υποχείλισε <-- 0
 max_στοιχεία <-- 0
 ΑρχήΕπανάληψης
  Γράψε '1. Ώθηση'
  Γράψε '2. Απώθηση'
  Γράψε '3. Πλήθος στοιχείων'
  Γράψε '4. Έξοδος'
  Διάβασε επ
  Αν επ = 1 τότε ! Ώθηση
   Αν top < 100 τότε
    top <-- top + 1
    Διάβασε Σ[top]
    Αν top = 100 τότε
     Π_γέμισε <-- Π_γέμισε + 1
    ΤέλοςΑν
    Αν top > max_στοιχεία τότε
     max_στοιχεία <-- top
    ΤέλοςΑν
   Αλλιώς
    Γράψε 'υπερχείλιση'
    Π_υπερχείλισε <-- Π_υπερχείλισε + 1
   ΤέλοςΑν
  ΑλλιώςΑν επ = 2 τότε ! Απώθηση
   Αν top <> 0 τότε
    Γράψε Σ[top]
    top <-- top-1
    Αν top = 0 τότε
     Π_άδειασε <-- Π_άδειασε + 1
    ΤέλοςΑν
   Αλλιώς
    Γράψε 'υποχείλιση'
    Π_υποχείλισε <-- Π_υποχείλισε + 1
   ΤέλοςΑν
  ΑλλιώςΑν επ = 3 τότε ! Πλήθος στοιχείων
   Γράψε top
  ΤέλοςΑν
 ΜέχριςΌτου επ = 4
 Γράψε Π_γέμισε, Π_άδειασε, Π_υπερχείλισε, Π_υποχείλισε,
max_στοιχεία
ΤέλοςΠρογράμματος
```

(§ 3.5) Ουρά

Ορισμός: δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα.



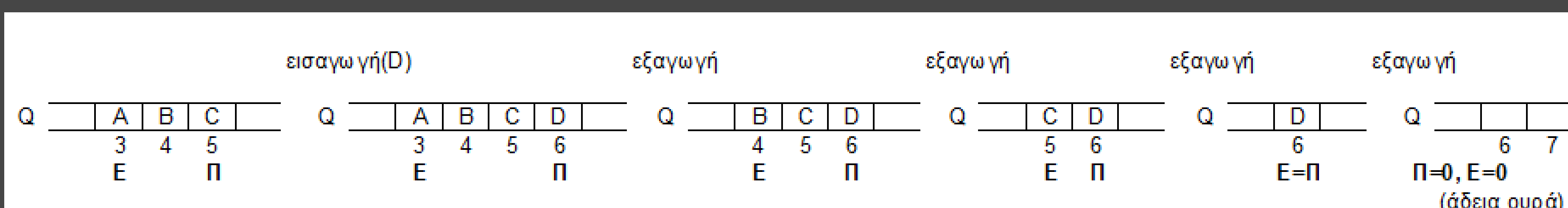
Σχ. 3.5 Υλοποίηση ουράς με χρήση πίνακα

Μέθοδος επεξεργασίας: «πρώτο μέσα, πρώτο έξω» (FIFO: first in – first out). Όπως για παράδειγμα: μία ουρά σε ταμείο, ή η ουρά ενός εκτυπωτή.

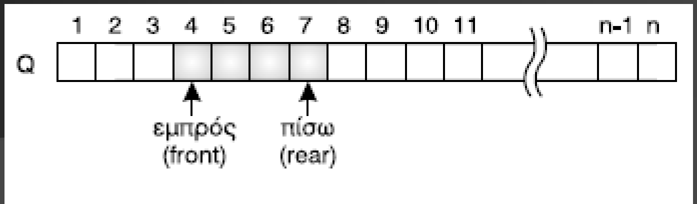
Κύριες λειτουργίες:

- **Εισαγωγή** (enqueue): ο rear αυξάνεται κατά 1 και το νέο στοιχείο εισάγεται στο Q[rear]. Έλεγχος για έλλειψη ελεύθερου χώρου ⇔ εισαγωγή σε γεμάτη ουρά
- **Εξαγωγή** (dequeue): εξάγεται το μπροστινό στοιχείο (Q[front]) και ο front αυξάνεται κατά 1. Έλεγχος για αποτυχημένη εξαγωγή ⇔ εξαγωγή σε άδεια ουρά

άδεια ουρά ⇔ front = rear = 0,
γεμάτη ουρά ⇔ rear = N



(§ 3.5) Ουρά (έστω ουρά Q[1000])



Σχ. 3.5 Υλοποίηση ουράς με χρήση πίνακα

Εισαγωγή στοιχείου x

Αν rear = 1000 τότε
  Γράψε 'γεμάτη ουρά'
ΑλλιώςΑν front=0 ΚΑΙ rear=0 τότε
  front ← 1
  rear ← 1
  Q[rear] ← x
Αλλιώς
  rear ← rear + 1
  Q[rear] ← x
ΤέλοςΑν

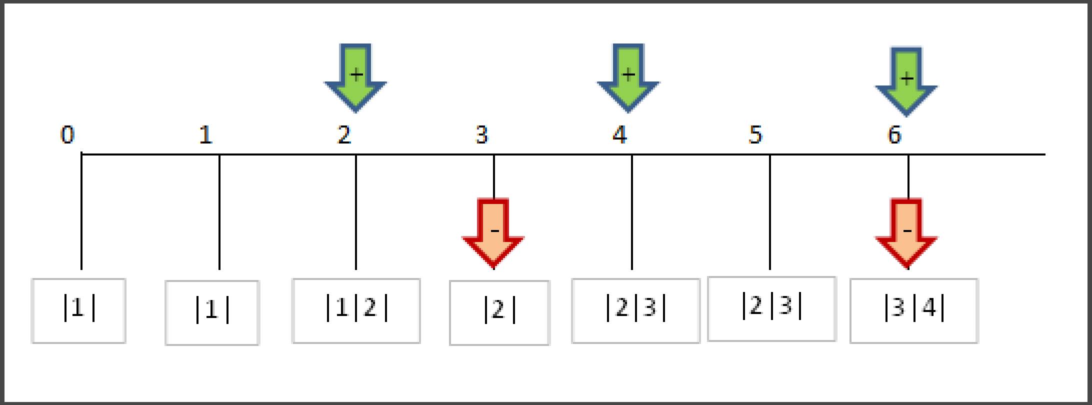
πλήθος στοιχείων ουράς
Αν front=0 ΚΑΙ rear=0 τότε
  πλήθος <-- 0
Αλλιώς
  πλήθος <-- rear - front + 1
ΤέλοςΑν

Εξαγωγή στοιχείου

Αν front=0 ΚΑΙ rear=0 τότε
  Γράψε 'άδεια ουρά'
ΑλλιώςΑν front = rear τότε
  Γράψε Q[front]
  front ← 0
  rear ← 0
Αλλιώς
  Γράψε Q[front]
  front ← front + 1
ΤέλοςΑν

(§ 3.5) Ουρά

Κατά την είσοδό τους σε μια τράπεζα οι πελάτες παίρνουν διαδοχικούς αριθμούς προτεραιότητας 1, 2, 3… που καθορίζουν τη σειρά τους στην ουρά του μοναδικού ταμείου. Κάθε 2 λεπτά της ώρας προσέρχεται ένας νέος πελάτης και προστίθεται στην ουρά. Ο ταμίας εξυπηρετεί κάθε φορά τον πρώτο πελάτη στην ουρά και η εξυπηρέτησή του διαρκεί 3 λεπτά ακριβώς. Μετά την εξυπηρέτησή του ο πελάτης αποχωρεί από την ουρά. Κατά την αρχή της διαδικασίας (χρόνος 0) στην ουρά υπάρχει μόνο ο πελάτης με αριθμό προτεραιότητας 1. Να γράψετε διαδοχικά, σε ξεχωριστές γραμμές, με τη σωστή σειρά, τους αριθμούς προτεραιότητας των πελατών που βρίσκονται στην ουρά του ταμείου αμέσως μετά το 1ο , 2ο , 3ο , 4ο , 5ο και 6ο λεπτό. (Πανελλαδικές 2016).



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;

1)

i)

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

front=1 και rear=5

ii) Η τελική μορφή της ουράς είναι:

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

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

	Ουρά 10 θέσεων				
	ΠΡΙΝ		λειτουργία	ΜΕΤΑ	
	front	rear		front	rear
1	0	0	εισαγωγή	1	1
2	0	0	εξαγωγή	0	0
3	4	4	εισαγωγή	4	5
4	4	4	εξαγωγή	0	0
5	5	7	εισαγωγή	5	8
6	5	7	εξαγωγή	6	7
7	7	10	εισαγωγή	7	10
8	7	10	εξαγωγή	8	10
9	1	10	εισαγωγή	1	10
10	1	10	εξαγωγή	2	10

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;

i) Η αρχική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
Ο	Σ	Λ	Τ	Ε					
front				**rear**					

front=1 και rear=5

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

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
			Τ	Ε	Λ	Ο	Σ		
			front				**rear**		

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

Άσκηση ουράς

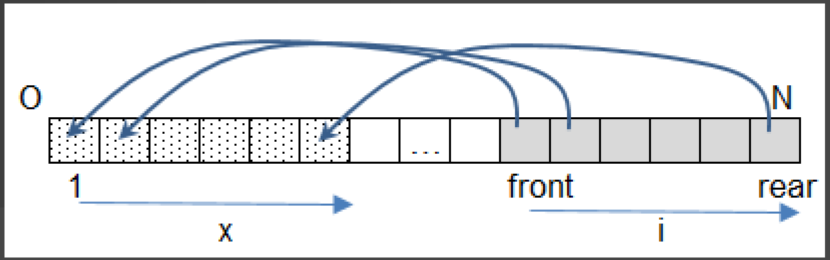
Να γραφεί πρόγραμμα διαχείρισης ακέραιας ουράς Q[100]. Αρχικά να την αδειάζει. Με επαναληπτικό μενού να εκτελεί τα παρακάτω: 1. Εισαγωγή 2. Εξαγωγή 3. Πλήθος στοιχείων 4. Έξοδος. Τελικά να εμφανίζει: α) πόσες φορές γέμισε και πόσες φορές άδειασε β) πόσες φορές απέτυχε η εισαγωγή και πόσες φορές απέτυχε η εξαγωγή γ) ποιο το μέγιστο πλήθος στοιχείων που είχε.

```
Πρόγραμμα Ουρά
  Μεταβλητές
    Ακέραιες: Q[100], front, rear, επ, Π_γέμισε,
Π_άδειασε, Π_απέτυχε_η_εισαγωγή,
Π_απέτυχε_η_εξαγωγή, max_στοιχεία, στοιχεία
Αρχή
  top <-- 0
  Π_γέμισε <-- 0
  Π_άδειασε <-- 0
  Π_απέτυχε_η_εισαγωγή <-- 0
  Π_απέτυχε_η_εξαγωγή <-- 0
  max_στοιχεία <-- 0
```

```
ΑρχήΕπανάληψης
   Γράψε '1. Εισαγωγή'
   Γράψε '2. Εξαγωγή'
   Γράψε '3. Πλήθος στοιχείων'
   Γράψε '4. Έξοδος'
   Διάβασε επ
   Αν επ = 1 τότε ! Εισαγωγή
     Αν rear = 100 τότε
       Γράψε 'γεμάτη ουρά'
       Π_απέτυχε_η_εισαγωγή <-- Π_απέτυχε_η_εισαγωγή + 1
     Αλλιώς
       Αν front=0 ΚΑΙ rear=0 τότε
         front <-- 1
         rear <-- 1
       Αλλιώς
         rear <-- rear + 1
       ΤέλοςΑν
       Διάβασε Q[rear]
       Αν rear = 100 τότε
         Π_γέμισε <-- Π_γέμισε + 1
       ΤέλοςΑν
       στοιχεία <-- rear - front + 1
       Αν στοιχεία > max_στοιχεία τότε
         max_στοιχεία <-- στοιχεία
       ΤέλοςΑν
     ΤέλοςΑν
```

```
 ΑλλιώςΑν επ = 2 τότε ! Εξαγωγή
      Αν front=0 ΚΑΙ rear=0 τότε
        Γράψε 'άδεια ουρά'
        Π_απέτυχε_η_εξαγωγή <--
Π_απέτυχε_η_εξαγωγή + 1
      Αλλιώς
        Γράψε Q[front]
        Αν front = rear τότε
          front <-- 0
          rear <-- 0
          Π_άδειασε <-- Π_άδειασε + 1
        Αλλιώς
          front <-- front + 1
        ΤέλοςΑν
      ΤέλοςΑν
    ΑλλιώςΑν επ = 3 τότε ! Πλήθος στοιχείων
      Αν front=0 ΚΑΙ rear=0 τότε
        πλήθος <-- 0
      Αλλιώς
        πλήθος <-- rear - front + 1
      ΤέλοςΑν
      Γράψε πλήθος
    ΤέλοςΑν
  ΜέχριςΌτου επ = 4
  Γράψε Π_γέμισε, Π_άδειασε,
Π_απέτυχε_η_εισαγωγή, Π_απέτυχε_η_εξαγωγή,
max_στοιχεία
ΤέλοςΠρογράμματος
```

Ολίσθηση ουράς
Τμήμα εντολών που σε ουρά (Ο[Ν], front, rear) και σε περίπτωση που ο rear έχει φτάσει στο δεξί της άκρο και έχει κενές θέσεις αριστερά του front, μετακινεί όλα τα στοιχεία της στην αρχή της ενημερώνοντας κατάλληλα και τους δείκτες της.




```
Αν rear=N ΚΑΙ front>1 τότε
  x <-- 0
  για i από front μέχρι rear
    x <-- x + 1
    O[x] <-- O[i]
  ΤέλοςΕπανάληψης
  front <-- 1
  rear <-- x
ΤέλοςΑν
```

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση της ουράς μπορεί να χρησιμοποιηθεί πίνακας.	❐	❐
2	Κατά την εισαγωγή ενός στοιχείου σε ουρά, αυτό τοποθετείται στο μπροστινό άκρο της.	❐	❐
3	Σε μια ουρά κάθε στοιχείο της εξάγεται από το μπροστινό άκρο της.	❐	❐
4	Η απώθηση είναι μια από τις λειτουργίες της ουράς.	❐	❐
5	Η εισαγωγή και η εξαγωγή είναι οι δύο βασικές λειτουργίες της ουράς.	❐	❐
6	Στην ουρά το στοιχείο που μπαίνει πρώτο βγαίνει και πρώτο.	❐	❐
7	Η υλοποίηση της ουράς χρησιμοποιεί μία μεταβλητή-δείκτη για την εκτέλεση των δύο βασικών λειτουργιών της.	❐	❐
8	Η λειτουργία της εξαγωγής μπορεί να εκτελεστεί σε μια γεμάτη ουρά.	❐	❐

Ε. 2: Δίνεται η επόμενη ακολουθία αριθμών : 4, 8, 2, 5, 9, 13. 1. Ποια λειτουργία θα χρησιμοποιήσετε για την τοποθέτηση των αριθμών σε ουρά; 2. Να σχεδιάσετε την ουρά έπειτα από την τοποθέτηση των αριθμών. 3. Ποια λειτουργία θα χρησιμοποιήσετε για την εξαγωγή των αριθμών από την ουρά; 4. Πόσες φορές θα πρέπει να εκτελεστεί η προηγούμενη λειτουργία στην ουρά για να εξαχθεί ο αριθμός 5;

Ε. 3: 1. Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Χ, Α, Β, Α, Ρ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα. i. Να προσδιορίσετε τις τιμές των δεικτών rear και front της παραπάνω ουράς και να τη σχεδιάσετε. ii. Αν εφαρμόσουμε τις ακόλουθες λειτουργίες: Εξαγωγή, Εξαγωγή, Εξαγωγή, Εισαγωγή Χ , Εισαγωγή Δ και Εξαγωγή ποιες είναι τις τιμές των δεικτών rear και front της παραπάνω ουράς και ποια η τελική μορφή της ουράς; 2. Σε μια κενή ουρά 10 θέσεων εισάγουμε τα στοιχεία Κ, Φ, Ι, Α,Ρ. Με ποιον τρόπο πρέπει να «εισαχθούν» και να «εξαχθούν» τα στοιχεία, ώστε να έχουμε ως έξοδο τα δεδομένα Α, Ρ, Χ, Η.

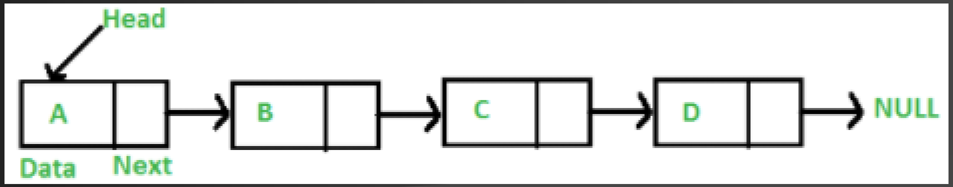
10 ωθήσεις σε άδεια πραγματική στοίβα Σ[100] με δείκτη top και με έλεγχο εγκυρότητας ώστε να είναι σε αύξουσα σειρά:

```
top <--- 1
Διάβασε Σ[top]
για i από 2 μέχρι 10
  ΑρχήΕπανάληψης
    Διάβασε x
  ΜέχριςΌτου x > Σ[top]
  top <-- top + 1
  Σ[top] <-- x
ΤέλοςΕπανάληψης
```

10 εισαγωγές σε άδεια πραγματική ουρά Ο[100] με δείκτες front και rear και με έλεγχο εγκυρότητας ώστε να είναι σε αύξουσα σειρά:

```
front <--- 1
rear <-- 1
Διάβασε Ο[rear]
για i από 2 μέχρι 10
  ΑρχήΕπανάληψης
    Διάβασε x
  ΜέχριςΌτου x > Ο[rear]
  rear <-- rear + 1
  Ο[rear] <-- x
ΤέλοςΕπανάληψης
```

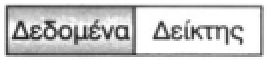
(§ 3.9) Άλλες δομές δεδομένων



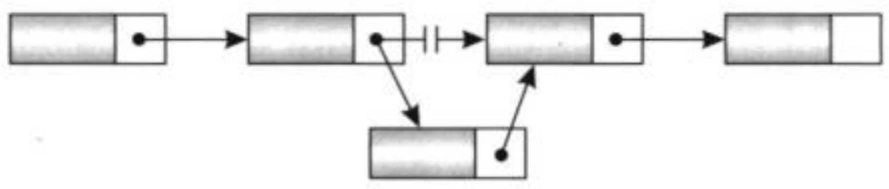
3.9.1 Λίστες

Οι κόμβοι τους βρίσκονται σε απομακρυσμένες θέσεις μνήμης και η σύνδεσή τους γίνεται με δείκτες.

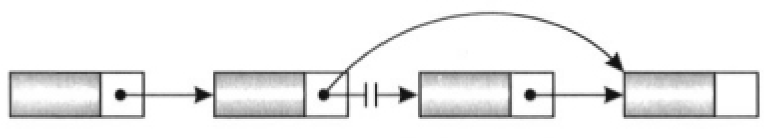
Δείκτης (pointer): ιδιαίτερος τύπος με τιμές που είναι διευθύνσεις στην κύρια μνήμη και χρησιμοποιείται για τη σύνδεση των κόμβων μιας δομής, που είναι αποθηκευμένοι σε μη συνεχόμενες θέσεις μνήμης.



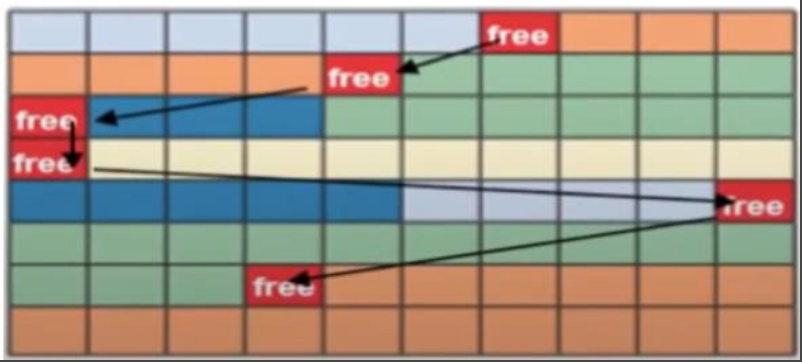
Σχ. 3.8 Δομή κόμβου λίστας

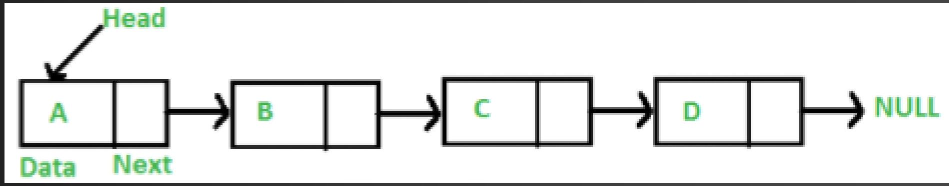


Σχ. 3.10. Εισαγωγή σε λίστα

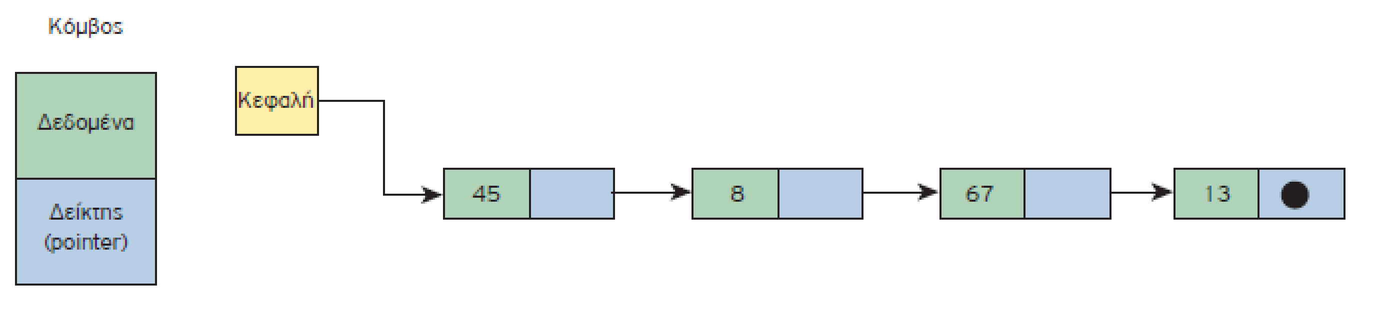


Σχ. 3.11. Διαγραφή κόμβου λίστας

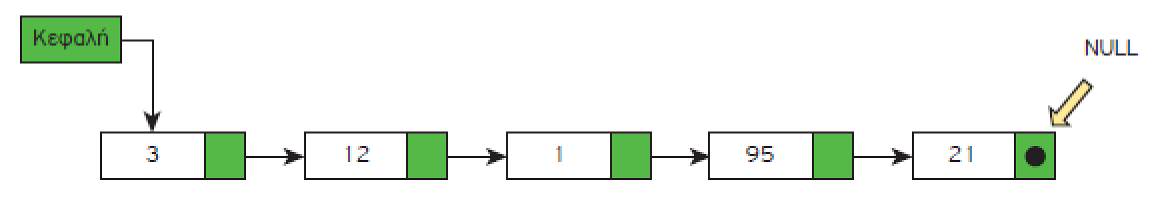




Μία (απλά) συνδεδεμένη λίστα (linked list) είναι ένα σύνολο κόμβων διατεταγμένων γραμμικά (ο ένας μετά τον άλλο). Κάθε κόμβος περιέχει εκτός από τα δεδομένα του και έναν δείκτη που δείχνει προς τον επόμενο κόμβο. Ο δείκτης του τελευταίου κόμβου δε δείχνει σε κάποιον κόμβο (δείκτης στο κενό). Για να το δηλώσουμε αυτό λέμε ότι το πεδίο δείκτη του τελευταίου κόμβου έχει την τιμή NULL. Για να προσπελάσουμε τους κόμβους της λίστας χρειάζεται να γνωρίζουμε τη διεύθυνση (θέση στη μνήμη) του πρώτου κόμβου της λίστας. Η διεύθυνση αυτή αποθηκεύεται σε μία ειδική μεταβλητή που την ονομάζουμε συνήθως Κεφαλή (Head).



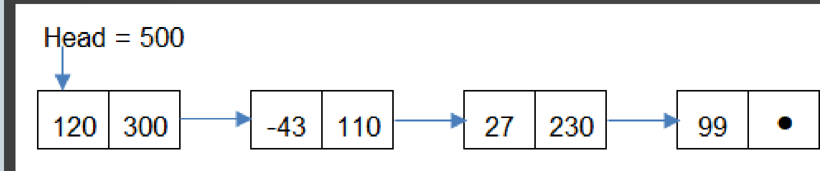
Εικόνα 1.3.2. α: Αναπαράσταση κόμβου β: Αναπαράσταση απλά συνδεδεμένης λίστας με 4 κόμβους

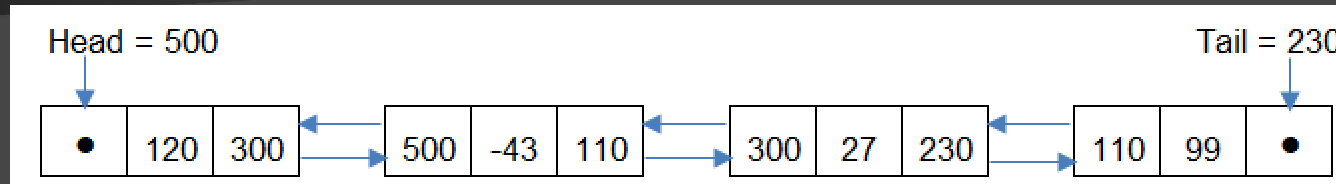


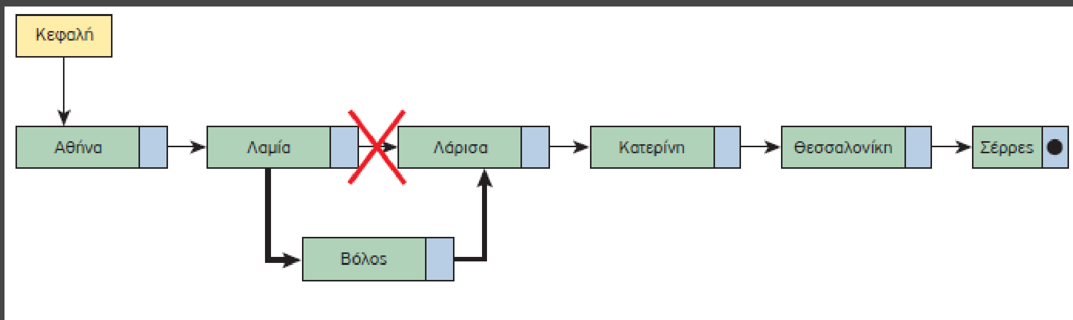
Εικόνα 1.3.3. Μία απλά συνδεδεμένη λίστα με 5 κόμβους.

Πρόσβαση στους κόμβους μιας συνδεδεμένης λίστας

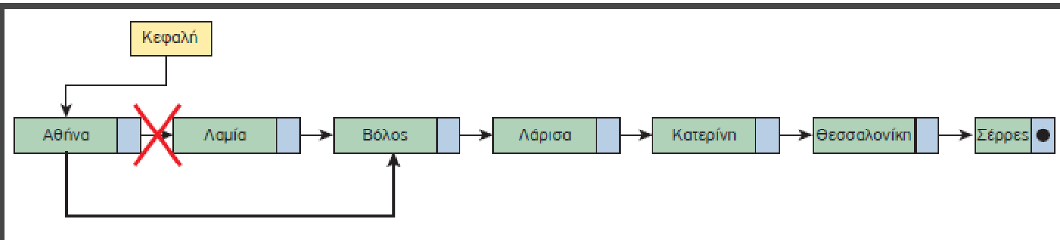
Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Αντίθετα, είναι διασκορπισμένοι σε όλη τη μνήμη και η σύνδεση μεταξύ τους γίνεται μέσω των δεικτών. Έχουμε άμεση πρόσβαση μόνο στον πρώτο κόμβο της λίστας. Επομένως, για να εντοπίσουμε κάποιον από τους ενδιάμεσους κόμβους, πρέπει να ξεκινήσουμε από τον πρώτο κόμβο της λίστας και να ακολουθήσουμε τους δείκτες με τη σειρά, μέχρι να φτάσουμε στον επιθυμητό κόμβο.







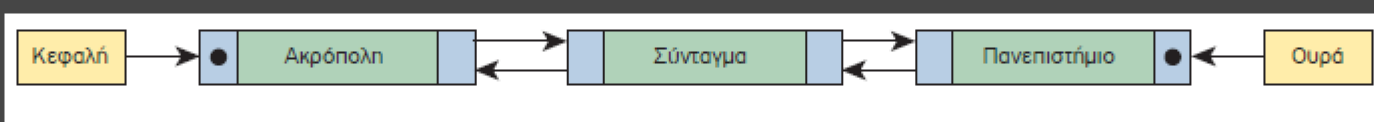
Εικόνα 1.3.5. Εισαγωγή νέου κόμβου στη λίστα



Εικόνα 1.3.6. Διαγραφή κόμβου από τη λίστα

Ποια η μορφή των παραπάνω λιστών μετά από κάθε μία από τις παρακάτω ενέργειες:

1. εισαγωγή στο τέλος της λίστας κόμβου με περιεχόμενο το 14 στη θέση μνήμης 400
2. εισαγωγή στην αρχή της λίστας κόμβου με περιεχόμενο το -33 στη θέση μνήμης 320
3. εισαγωγή ως 3ου κόμβου με περιεχόμενο το 77 στη θέση μνήμης 620
4. διαγραφή του προτελευταίου κόμβου
5. εγγραφή της τιμής -17 στη θέση μνήμης 110



Εικόνα 1.3.8. Οι σταθμοί του μετρό - Μια διπλά συνδεδεμένη λίστα

3.9.1 Λίστες – Άσκηση (Πανελλαδικές 2016 επαναληπτικές)

Δίνεται μια λίστα η οποία αποτελείται από 5 κόμβους. Το πρώτο πεδίο του κάθε κόμβου είναι ένα γράμμα και το δεύτερο πεδίο είναι η διεύθυνση του επόμενου κόμβου, όπως φαίνεται στο παρακάτω διάγραμμα, που σχηματίζει τη λέξη ΔΕΚΤΗ:

Η λίστα αυτή απεικονίζεται στη μνήμη με τη μορφή που φαίνεται στο παρακάτω σχήμα.

...	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	...
...		Ε	25		Δ	16					Κ	30		Η	0	Τ	28	...

Στον τελευταίο κόμβο, το δεύτερο πεδίο έχει την τιμή 0, που σημαίνει το τέλος της λίστας.
α. Να σχεδιάσετε στο τετράδιό σας την απεικόνιση της μνήμης μετά από τη διαγραφή του κατάλληλου κόμβου από την αρχική λίστα, ώστε να σχηματιστεί η λέξη ΔΕΤΗ. (μονάδες 2)

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
	Ε	30		Δ	16					Κ	0		Η	0	Τ	28	

β. Να σχεδιάσετε στο τετράδιό σας την απεικόνιση της μνήμης μετά από την εισαγωγή, στην αρχική λίστα, του κόμβου με πρώτο πεδίο το γράμμα Α στη θέση 21, ώστε να σχηματιστεί η λέξη ΔΕΚΑΤΗ. (μονάδες 4)

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
	Ε	25		Δ	16	Α	30			Κ	21		Η	0	Τ	28	

Διαφορές Λίστας σε σχέση με τον Πίνακα

1. πίνακας: δομή τυχαίας προσπέλασης, λίστα: δομή ακολουθιακής ή σειριακής προσπέλασης. Για να φθάσουμε, δηλαδή, σ' έναν κόμβο μιας λίστας πρέπει να περάσουμε από όλους τους προηγούμενους ξεκινώντας από τον πρώτο.
2. πίνακας: σταθερό μέγεθος, το οποίο δηλώνεται εξαρχής κατά την υλοποίηση (στατική δομή δεδομένων), λίστα: δυναμική δομή και το μέγεθός της μπορεί να μεταβάλλεται καθώς εισέρχονται νέοι κόμβοι στη λίστα ή διαγράφονται κάποιοι άλλοι.
3. πίνακας: τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις μνήμης, λίστα: Οι κόμβοι αποθηκεύονται σε μη συνεχόμενες θέσεις μνήμης.

Πλεονεκτήματα

1. δυναμικό μέγεθος
2. ευκολία εισαγωγής και διαγραφής από οποιοδήποτε μέρος της λίστας
3. μη αναγκαιότητα δήλωσης του μεγέθους τους

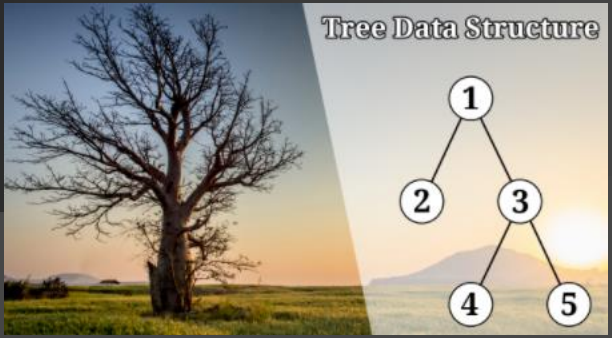
Μειονεκτήματα

1. Η τυχαία πρόσβαση στη λίστα δεν επιτρέπεται. Είναι αδύνατο να φτάσετε στον n-οστό κόμβο μιας απλά συνδεδεμένης λίστας χωρίς πρώτα να περάσετε από όλους τους κόμβους διαδοχικά μέχρι τον συγκεκριμένο κόμβο ξεκινώντας από τον πρώτο κόμβο. Εναλλακτικά, στην περίπτωση της διπλά συνδεμένης λίστας μπορείτε να ξεκινήσετε και από τον τελευταίο κόμβο. Επομένως, δεν μπορούμε να πραγματοποιήσουμε με αποτελεσματικό τρόπο δυαδική αναζήτηση σε συνδεδεμένες λίστες.
2. Οι συνδεδεμένες λίστες έχουν πολύ μεγαλύτερη επιβάρυνση από τους πίνακες, αφού οι συνδεδεμένοι κόμβοι της λίστας είναι δυναμικά κατανεμημένοι (οι οποίοι είναι λιγότερο αποτελεσματικοί στη χρήση της μνήμης) και κάθε κόμβος στη λίστα πρέπει, επιπλέον, να αποθηκεύσει έναν πρόσθετο δείκτη που θα δείχνει στον επόμενο κόμβο. Στην περίπτωση των διπλά συνδεδεμένων λιστών χρειαζόμαστε επιπλέον έναν δεύτερο δείκτη που θα δείχνει στον προηγούμενο κόμβο.

Βασικές πράξεις

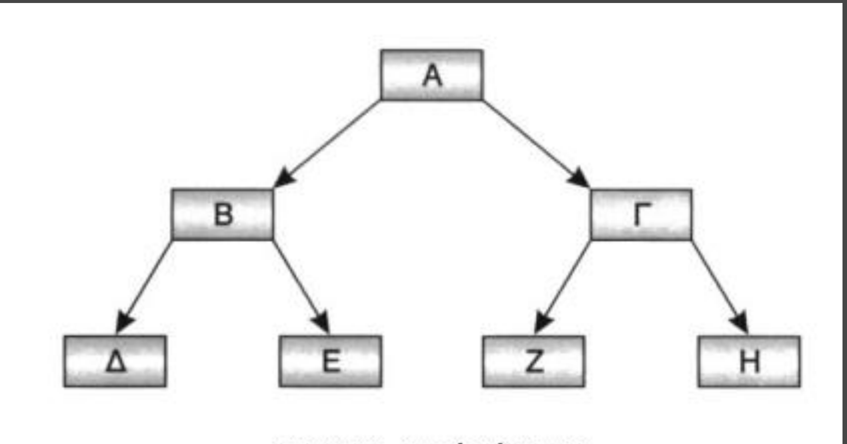
1. Εισαγωγή κόμβου
2. Διαγραφή κόμβου
3. Έλεγχος για το αν η λίστα είναι κενή.
4. Αναζήτηση κόμβου για την εύρεση συγκεκριμένου στοιχείου.
5. Διάσχιση της λίστας και προσπέλαση των στοιχείων της

(§ 3.9) Άλλες δομές δεδομένων

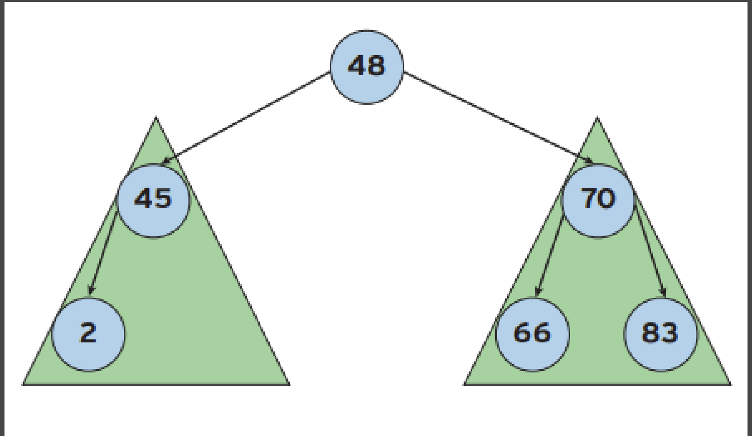


3.9.2 Δέντρα

Δένδρα (trees): δομές που υλοποιούνται με τη βοήθεια των δεικτών. Από ένα κόμβο δεν υπάρχει ένας μόνο επόμενος κόμβος, αλλά περισσότεροι. Υπάρχει ένας μόνο κόμβος, που λέγεται ρίζα, από τον οποίο ξεκινούν όλοι οι άλλοι κόμβοι.

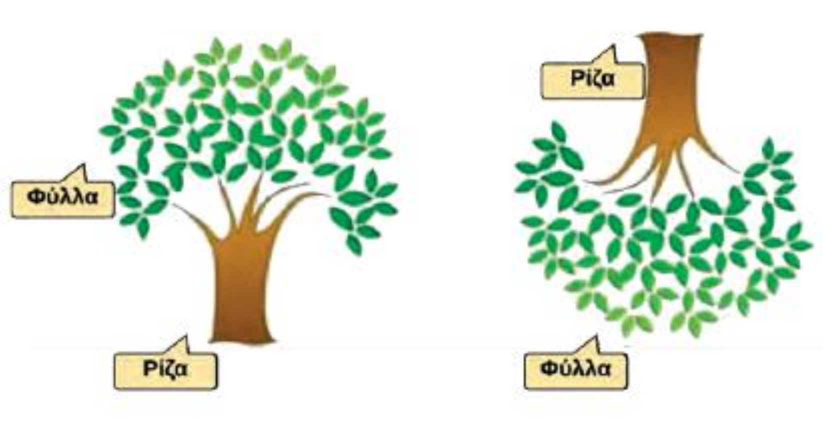


Σχ. 3.12. Δομή δένδρου

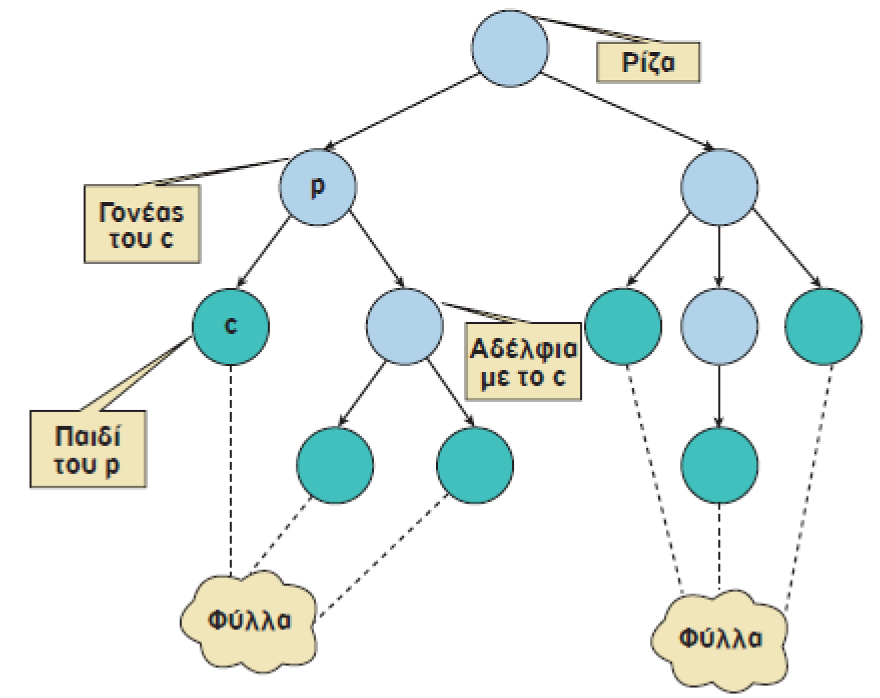


Εικόνα 1.3.14. Αριστερό και δεξιό υποδένδρο του κόμβου 48

Ρίζα: Α, Παιδιά του Β: Δ και Ε, Πρόγονος του Β: Α, Φύλα: Δ,Ε,Ζ,Η Μονοπάτι του Ε: Α-Β-Ε



Εικόνα 1.3.9. Τα δένδρα στη φύση και στην Πληροφορική

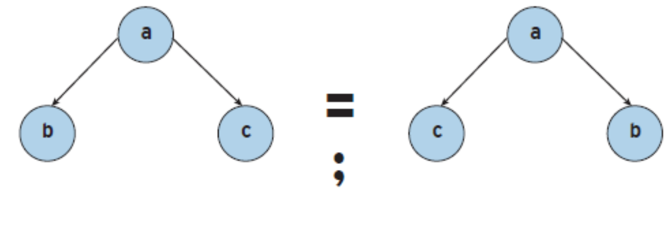


Ένα **δένδρο (tree)** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων και ένα σύνολο ακμών μεταξύ των κόμβων με βάση τους εξής κανόνες:

- Υπάρχει ένας ξεχωριστός κόμβος που ονομάζεται ρίζα. Αυτός είναι ένας κόμβος χωρίς γονέα.
- Για κάθε κόμβο c, εκτός από τη ρίζα, υπάρχει μόνο μια ακμή που καταλήγει στον κόμβο αυτόν ξεκινώντας από κάποιον άλλον κόμβο p. Ο κόμβος p ονομάζεται γονέας του c και ο κόμβος c παιδί του p.
- Για κάθε κόμβο υπάρχει μία μοναδική διαδρομή, δηλαδή, μια ακολουθία διαδοχικών ακμών, που ξεκινάει από τη ρίζα και τερματίζει σε αυτόν τον κόμβο.

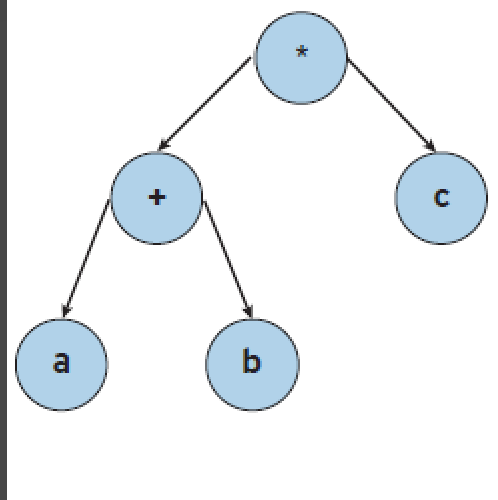
Δένδρο θεωρούμε και το κενό δένδρο, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

Αν για κάθε κόμβο υπάρχει μία γραμμική σχέση μεταξύ των παιδιών του κόμβου αυτού, αναφερόμαστε σε ένα **διατεταγμένο δένδρο**



Εικόνα 1.3.15. Πότε δύο δένδρα είναι ίδια

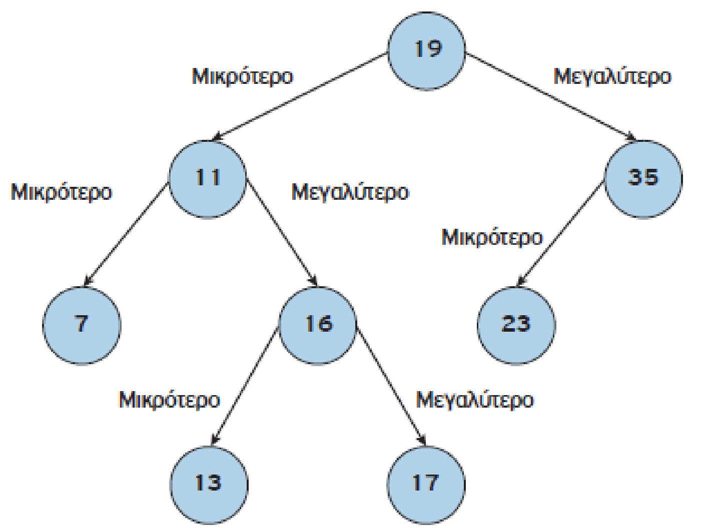
Ένα **δυαδικό δένδρο** είναι ένα διατεταγμένο δένδρο, στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, το αριστερό και το δεξί παιδί.



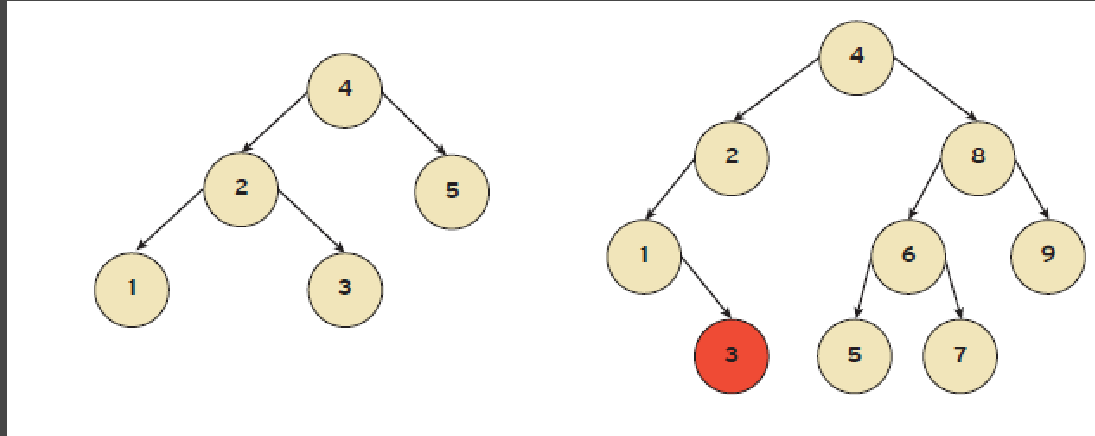
Εικόνα 1.3.20. (a+b)*c

Δυαδικά Δένδρα Αναζήτησης

Ένα **δυαδικό δένδρο αναζήτησης** (binary search tree) είναι ένα δυαδικό δένδρο, όπου για κάθε κόμβο u, όλοι οι κόμβοι του αριστερού υποδένδρου έχουν τιμές μικρότερες της τιμής του κόμβου u και όλοι οι κόμβοι του δεξιού υποδένδρου έχουν τιμές μεγαλύτερες (ή ίσες) της τιμής του κόμβου u. Για λόγους απλούστευσης θεωρούμε ότι δεν υπάρχουν τιμές ίσες με την τιμή του κόμβου u.

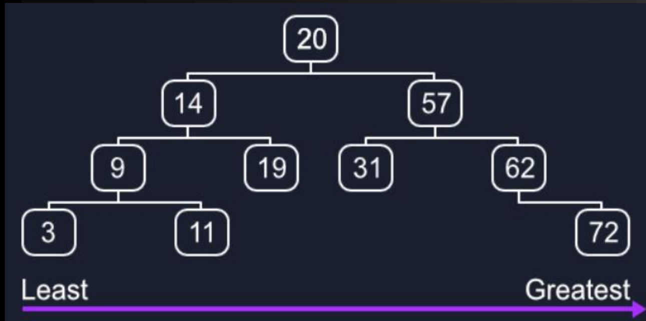


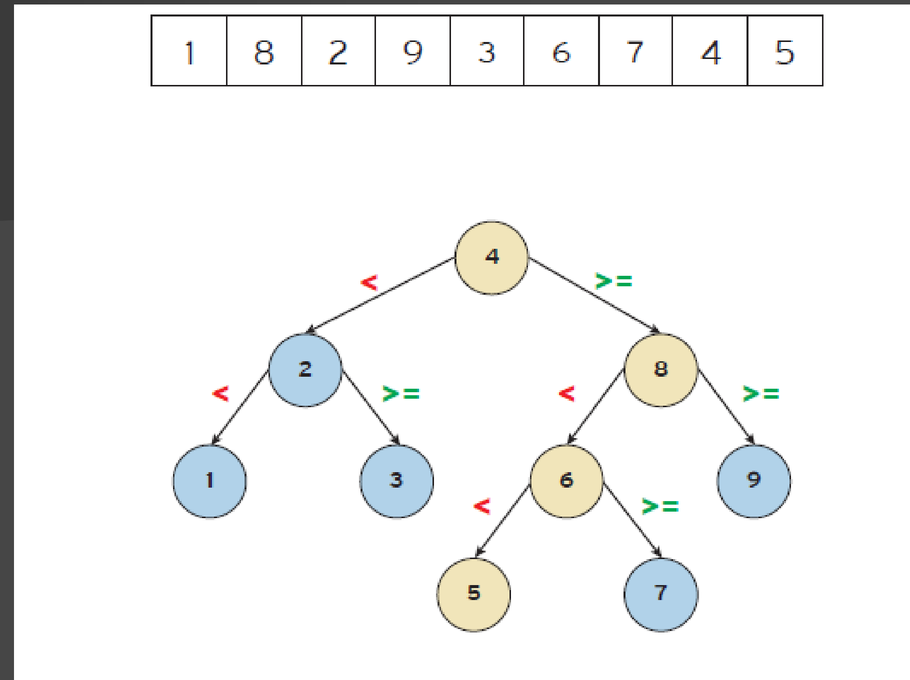
Εικόνα 1.3.22. Δυαδικό δένδρο αναζήτησης



Εικόνα 1.3.23. α: Δυαδικό δένδρο αναζήτησης β: Μη δυαδικό δένδρο αναζήτησης

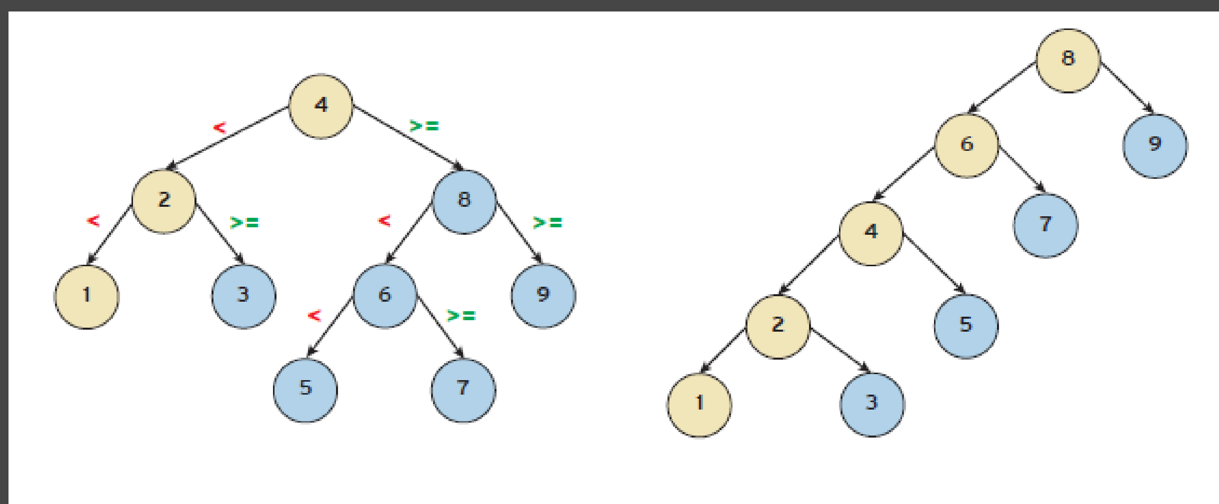
Η αναζήτηση για μια συγκεκριμένη τιμή γίνεται ταχύτερα χάρη στον τρόπο αποθήκευσης των τιμών.



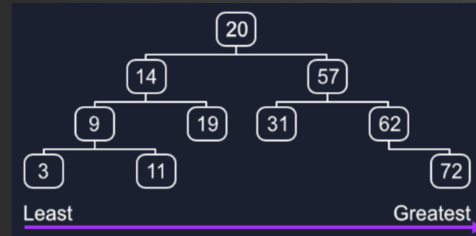


Εικόνα 1.3.24. Αλγόριθμος αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης

Το πρώτο δένδρο είναι πιο «ισορροπημένο» σε σχέση με το δεύτερο

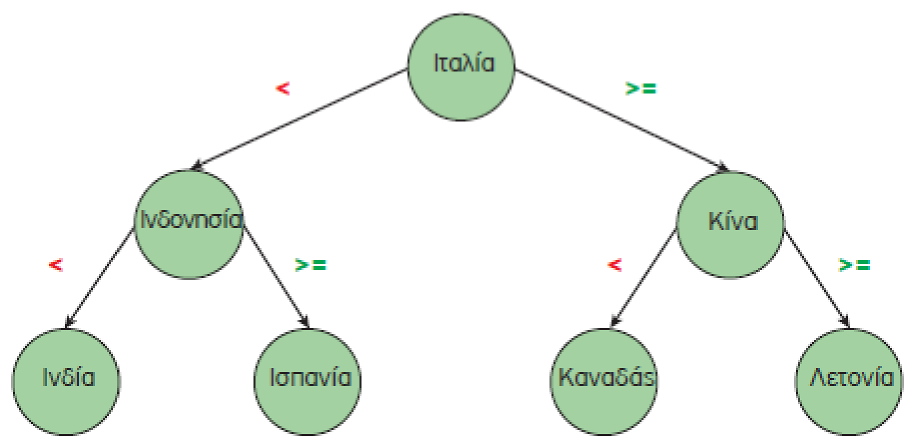


Εικόνα 1.3.25. Η σημασία της δομής ενός δυαδικού δένδρου αναζήτησης στην εύρεση στοιχείων



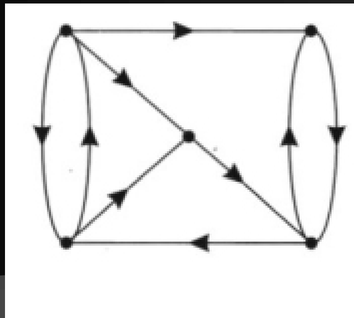
Τα δυαδικά δένδρα αναζήτησης συνδυάζουν τα πλεονεκτήματα των λιστών, όσον αφορά τις πράξεις της εισαγωγής και της διαγραφής, αλλά και τα πλεονεκτήματα των ταξινομημένων πινάκων, όσον αφορά την πράξη της αναζήτησης

Ινδία	Ινδονησία	Ισπανία	Ιταλία	Καναδάς	Κίνα	Λετονία



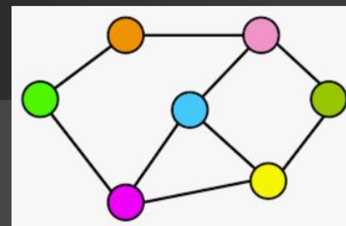
Εικόνα 1.3.26. Αναζήτηση σε ταξινομημένο πίνακα και σε δυαδικό δένδρο αναζήτησης

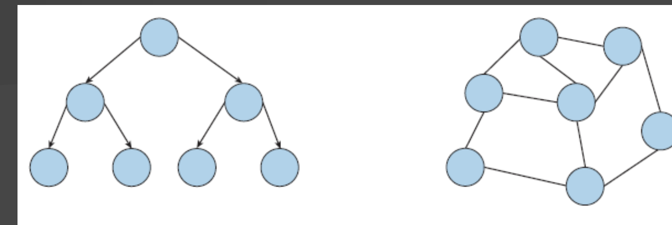
3.9.3 Γράφοι



Σχ. 3.13. Ένας γράφος

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





Εικόνα 1.3.27.

α: Δένδρο β: Γράφος

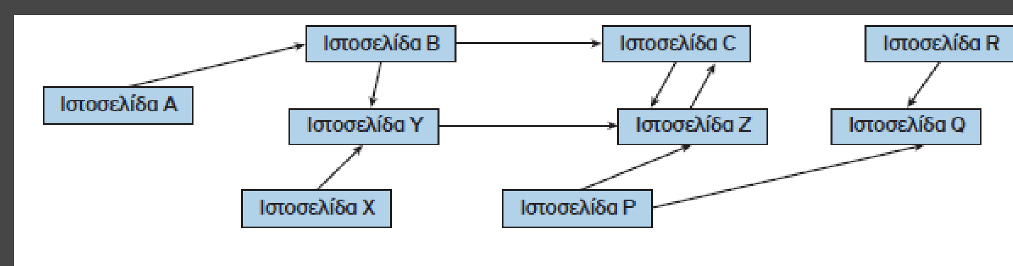
Τύποι Γράφων
1. κατευθυνόμενοι
2. μη κατευθυνόμενοι

α: Κατευθυνόμενη ακμή β: μη κατευθυνόμενη ακμή

Σε μια κατευθυνόμενη ακμή, μπορούμε να ταξιδέψουμε μόνο από την προέλευση στον προορισμό. Σε μια μη κατευθυνόμενη ακμή, η διαδρομή μεταξύ των δύο κόμβων είναι αμφίδρομη.

Εάν όλες οι ακμές σε έναν γράφο έχουν κατεύθυνση, ο γράφος ονομάζεται **κατευθυνόμενος γράφος (directed graph)**.
Εάν όλες οι ακμές σε έναν γράφο δεν έχουν κατεύθυνση, ο γράφος ονομάζεται **μη κατευθυνόμενος γράφος (undirected graph).**

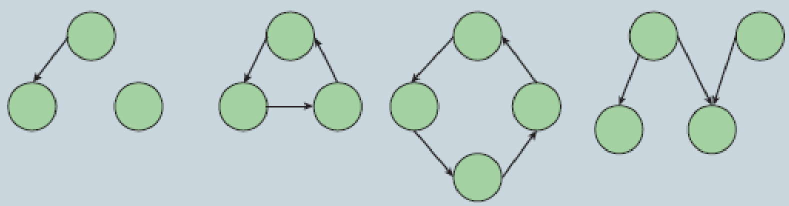
Μέγιστος αριθμός ακμών σε μη κατευθυνόμενο γράφο με Ν κόμβους = N*(N-1)/2



Εικόνα 1.3.29. Παγκόσμιος Ιστός – Η ιστοσελίδα R περιέχει σύνδεσμο προς την ιστοσελίδα Q

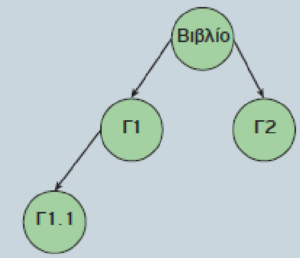
α/α	Προτάσεις	Σ	Λ
1	Μια απλά συνδεδεμένη λίστα μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις.	❐	❐
2	Σε μία λίστα δε χρειάζεται να οριστεί ένα αρχικό μέγεθος.	❐	❐
3	Δεν είναι δυνατό να υπάρχει «τυχαία» πρόσβαση σε μια απλά συνδεδεμένη λίστα.	❐	❐
4	Σε μια λίστα, τα στοιχεία δεν μπορούν να προστεθούν ή να αφαιρεθούν από τη μέση της λίστας, παρά μόνο από την αρχή ή το τέλος της.	❐	❐
5	Στη διπλά συνδεδεμένη λίστα τα περιεχόμενα των κόμβων προσπελαύνονται και από τις δύο κατευθύνσεις.	❐	❐

Δένδρα ή Γράφοι;
Στην Εικόνα 1.3.31 ποιες από τις παρακάτω δομές είναι δένδρα και ποιες είναι γράφοι. Προσπαθήστε να εξηγήσετε το γιατί.



α/α	Προτάσεις	Σ	Λ
1	Η ρίζα ενός δένδρου δεν μπορεί ποτέ να είναι φύλλο.	❐	❐
2	Σε ένα δυαδικό δένδρο, φύλλα συναντάμε μόνο στο αριστερό υποδένδρο.	❐	❐
3	Σε ένα δυαδικό δένδρο, κάθε κόμβος-γονέας μπορεί να έχει το πολύ δύο παιδιά	❐	❐
4	Δεν είναι δυνατό να υπάρχουν δύο διαφορετικές διαδρομές από την ρίζα προς έναν άλλον κόμβο ενός δένδρου.	❐	❐
5	Σε ένα δυαδικό δένδρο, κάθε κόμβος έχει μηδέν, ένα ή δύο υποδένδρα.	❐	❐
6	Η ρίζα ενός δένδρου είναι ο μόνος κόμβος ενός δένδρου που δεν έχει γονέα.	❐	❐
7	Τα φύλλα ενός δένδρου είναι απομονωμένοι κόμβοι που δε συνδέονται με άλλους κόμβους.	❐	❐
8	Σε ένα δένδρο, κάθε κόμβος-γονέας μπορεί να έχει οποιονδήποτε αριθμό παιδιών	❐	❐
9	Μπορούν να υπάρχουν διαφορετικές δομές δυαδικών δένδρων αναζήτησης που αποθηκεύουν τα ίδια στοιχεία.	❐	❐
10	Κάθε δένδρο είναι γράφος	❐	❐

Βιβλίο
- Γ1
  - Γ1.1
  - Γ1.2
- Γ2
  - Γ2.1
    - Γ2.1.1
    - Γ2.1.2
  - Γ2.2
  - Γ2.3
- Γ3



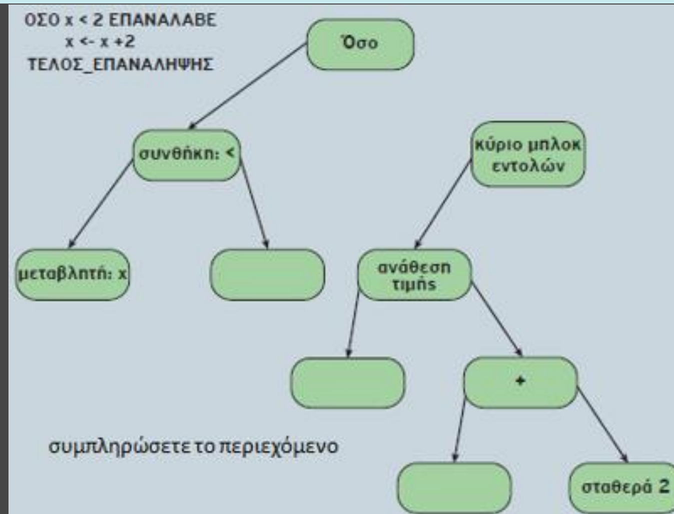
συμπληρώσετε το δένδρο

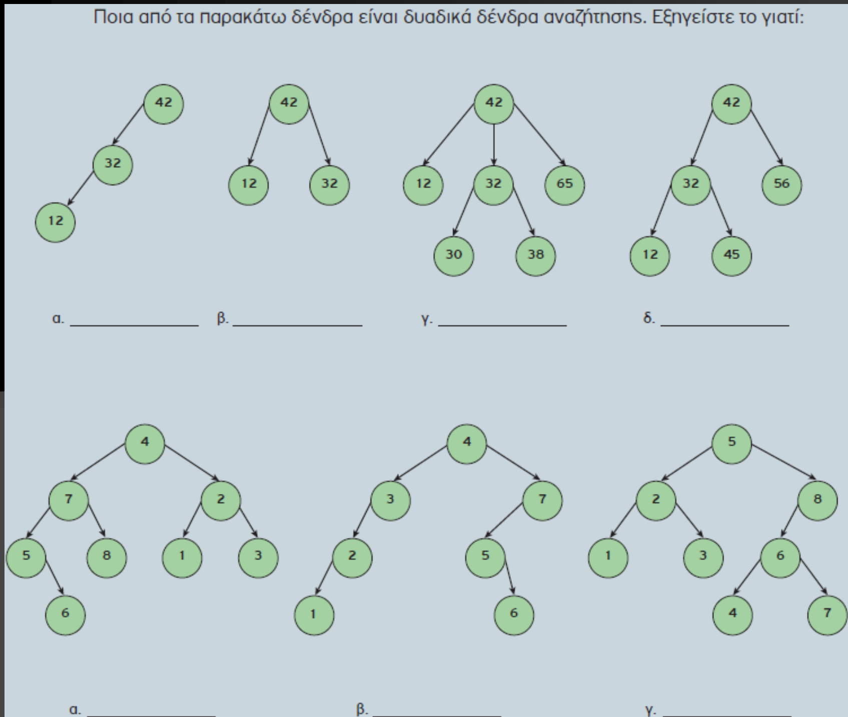
1. Ποια από τις βασικές δομές δεδομένων είναι η πιο κατάλληλη για να αναπαραστήσετε τη δομή των καταλόγων, των υποκαταλόγων και των αρχείων στον σκληρό σας δίσκο:

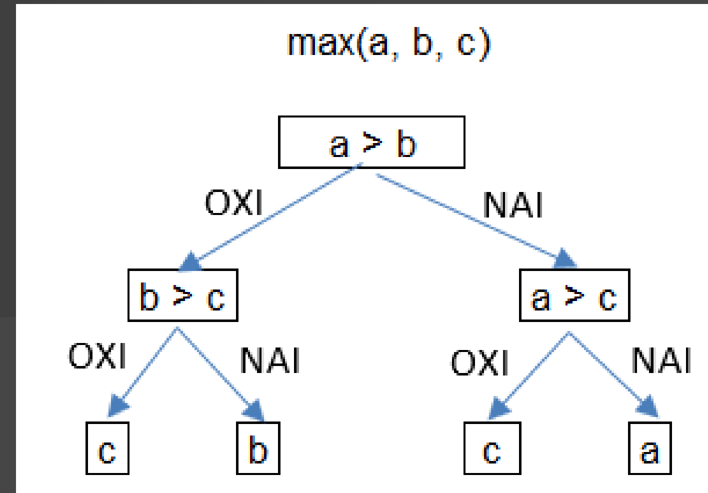
- ❐ πίνακας
- ❐ λίστα
- ❐ δένδρο
- ❐ ουρά
- ❐ στοίβα

2. Ο κατάλογος των φοιτητών που εγγράφονται σε ένα μάθημα είναι ταξινομημένος αλφαβητικά με βάση το ονοματεπώνυμο και περιλαμβάνει ένα σύνολο πληροφοριών σχετικών με τον φοιτητή, όπως είναι ο κωδικός του φοιτητή, η ημερομηνία γέννησης, το φύλο, η διεύθυνση, ο αριθμός τηλεφώνου κ.λπ. Επιλέξτε ποια από τις παρακάτω δομές δεδομένων είναι καταλληλότερη για την αναπαράσταση αυτών των πληροφοριών:

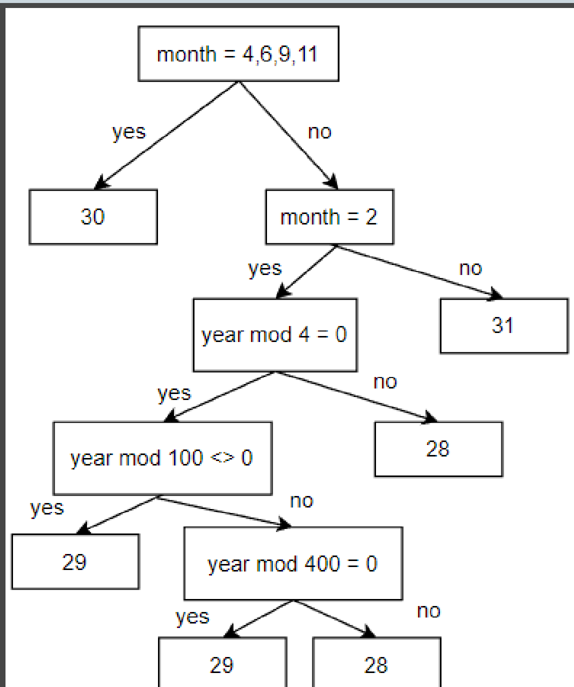
- ❐ στοίβα
- ❐ δένδρο
- ❐ λίστα
- ❐ ουρά



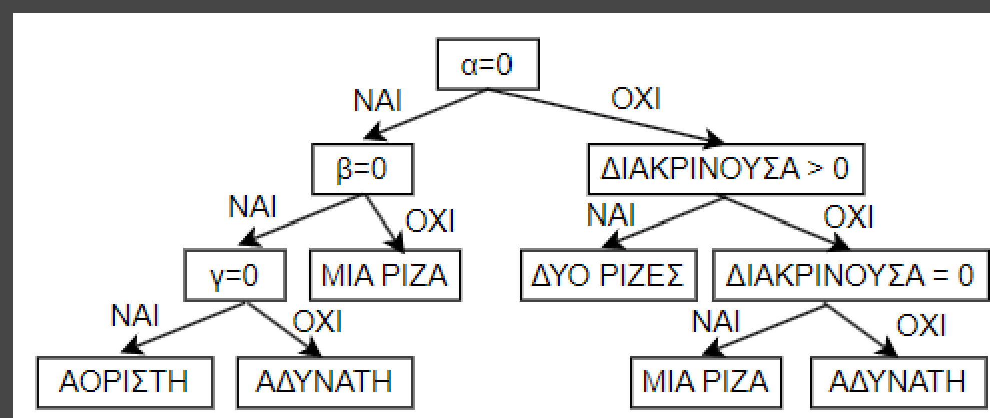




Στα **δένδρα απόφασης** κάθε κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ιδιότητα), κάθε ακμή αντιπροσωπεύει μια απόφαση (κανόνα) και κάθε φύλλο αντιπροσωπεύει ένα αποτέλεσμα. Στους αλγορίθμους μηχανικής μάθησης (machine learning) τα δένδρα απόφασης έχουν πρωτεύοντα ρόλο.

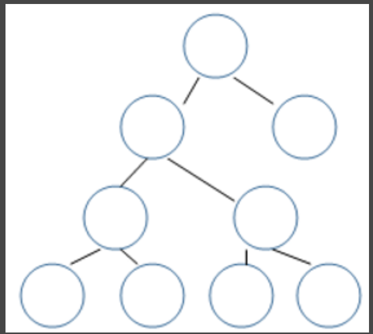


Δευτεροβάθμια $\alpha x^2+\beta x+\gamma=0$



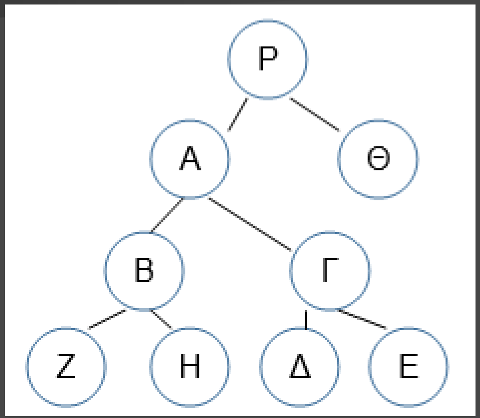
Γραμμική δομή: μετά από κάθε στοιχείο ακολουθεί ένα άλλο στοιχείο εκτός και αν είναι το τελευταίο. Γραμμικές δομές: πίνακας, στοίβα, ουρά, λίστα. Μη γραμμικές δομές: δέντρο, γράφος.

Τοποθετήστε τους αριθμούς 1-9 στο παρακάτω δέντρο ώστε να είναι δυαδικό δέντρο αναζήτησης

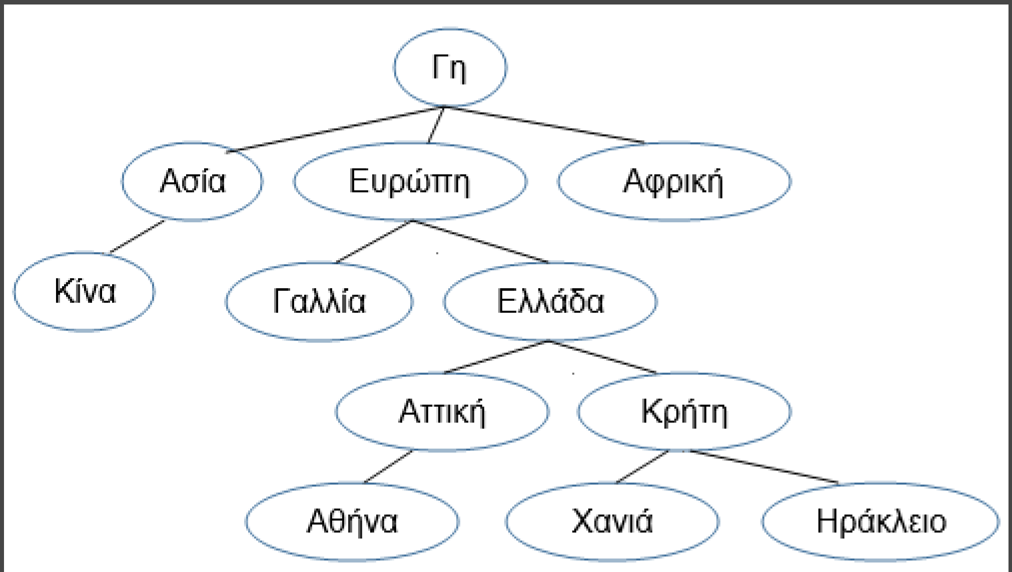


Να σχεδιασθεί το δέντρο με τις εξής σχέσεις:

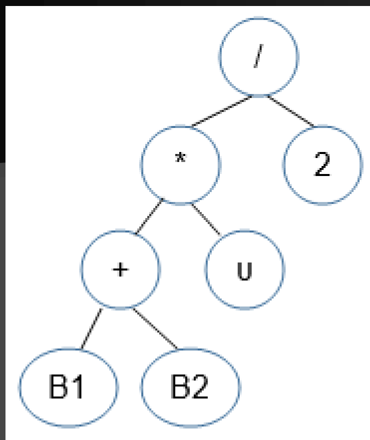
1. το Α είναι πρόγονος του Β
2. το Δ είναι απόγονος του Γ
3. τα Β και Γ είναι αδέρφια
4. τα Ε και Δ είναι αδέρφια
5. το Β είναι πρόγονος του Η
6. το Θ είναι αδερφός του Α
7. το Ρ είναι ρίζα
8. το μονοπάτι του Ζ είναι: Ρ-Α-Β-Ζ



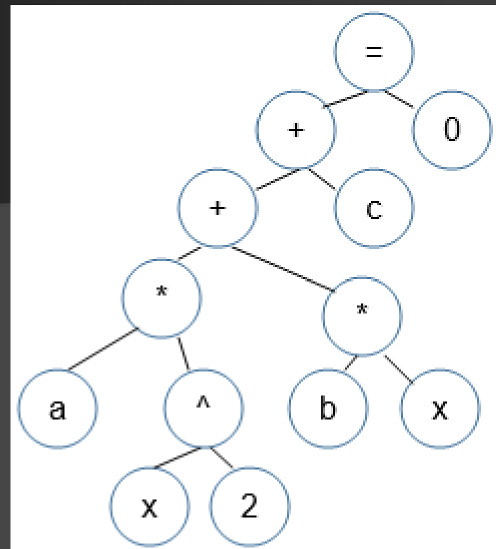
Να σχεδιασθεί το δέντρο των εξής γεωγραφικών περιοχών: Ελλάδα, Ασία, Γαλλία, Χανιά, Αφρική, Αθήνα, Ευρώπη, Ηράκλειο, Γη, Αττική, Κίνα, Κρήτη



Να σχεδιασθεί το δέντρο της έκφρασης: (B1+B2)*u/2



Να σχεδιασθεί το δέντρο της έκφρασης: a*x^2+b*x+c = 0

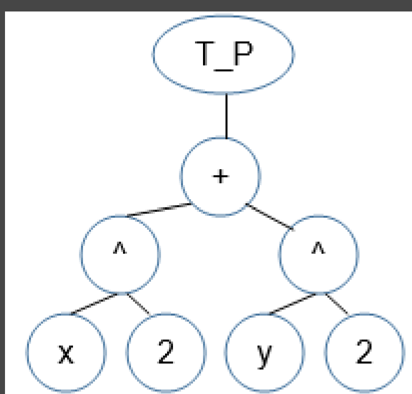


Συμπληρώστε τα κενά ώστε να προκύψει λίστα που να σχηματίζει τη λέξη "ΛΥΚΕΙΟ"

Θέση μνήμης	Κόμβος	
	Τιμή	Δείκτης
		Κεφαλή: ...
4	Κ	...
50	Λ	...
80	Υ	...
100	Ο	...
110	Ε	...
200	Ι	...

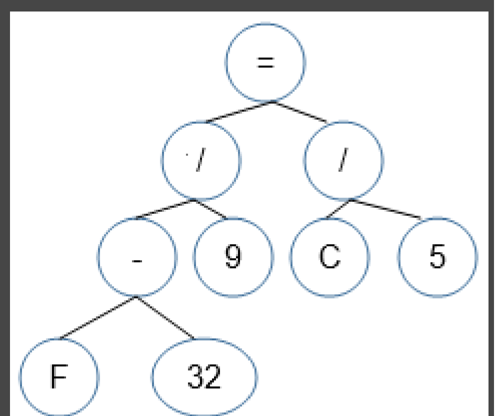
Απλά συνδεδεμένη

Να γραφεί η έκφραση που ισοδυναμεί με το δέντρο:



T_P(x^2+y^2)

Να γραφεί η έκφραση που ισοδυναμεί με το δέντρο:



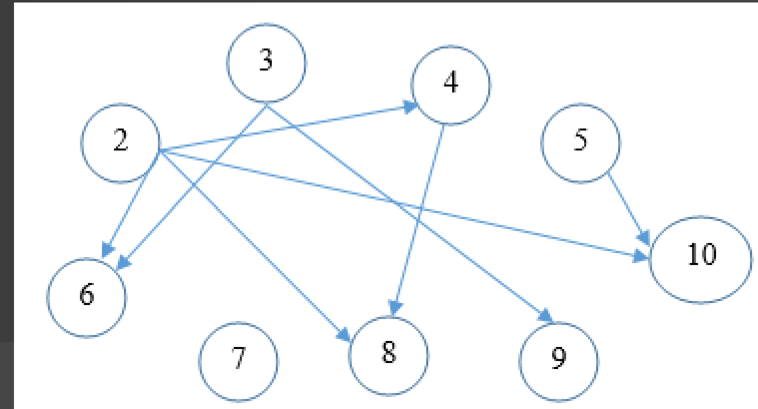
(F-32)/9=C/5

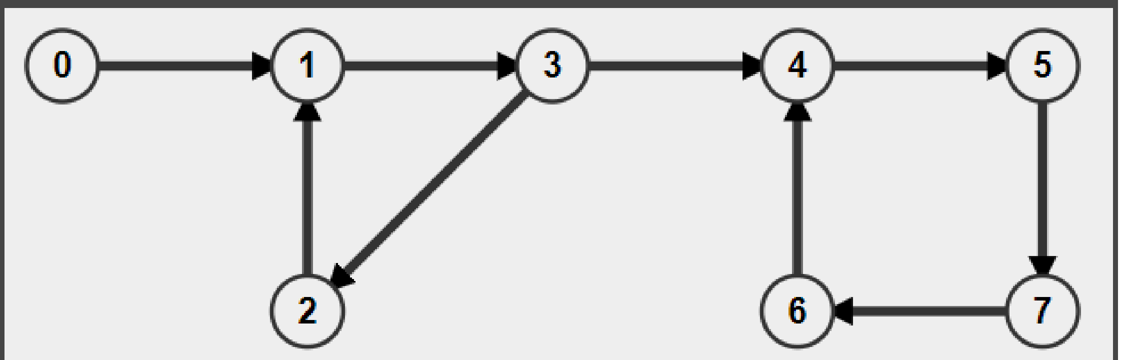
Κεφαλή	...
Ουρά	...

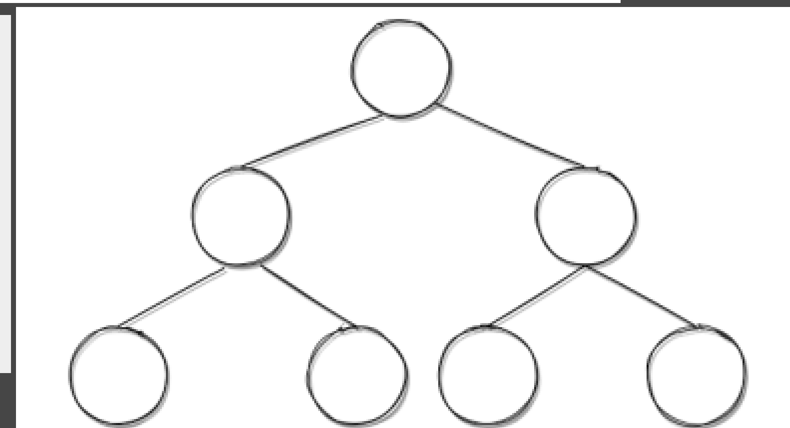
Θέση μνήμης	Κόμβος		
	Τιμή	Δείκτης Προηγούμενος	Δείκτης Επόμενος
4	Κ	...	...
50	Λ	...	...
80	Υ	...	...
100	Ο	...	...
110	Ε	...	...
200	Ι	...	...

Διπλά συνδεδεμένη

Για τους ακέραιους αριθμούς από το 2 έως και το 10 να σχεδιασθεί ο κατευθυνόμενος γράφος που περιέχει τις ακμές: από το α στο β εάν α < β και ο α είναι διαιρέτης του β. π.χ. 3 → 9





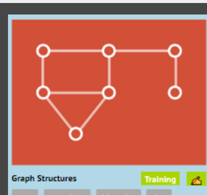


Πίνακας Γειτνίασης **Adjacency Matrix**

	0	1	2	3	4	5	6	7
0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0
4	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	1
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0

Εισάγετε τις τιμές: 92, -5, 45, 0, 16, 6 και 9 στους κόμβους του παραπάνω δέντρου ώστε να είναι δυαδικό δέντρο αναζήτησης

Σχεδιάστε το δυαδικό δέντρο αναζήτησης που θα προκύψει εάν προσθέσουμε σε αυτό τις τιμές: 4,2,1,6,5,7,3 και με αυτή τη σειρά.

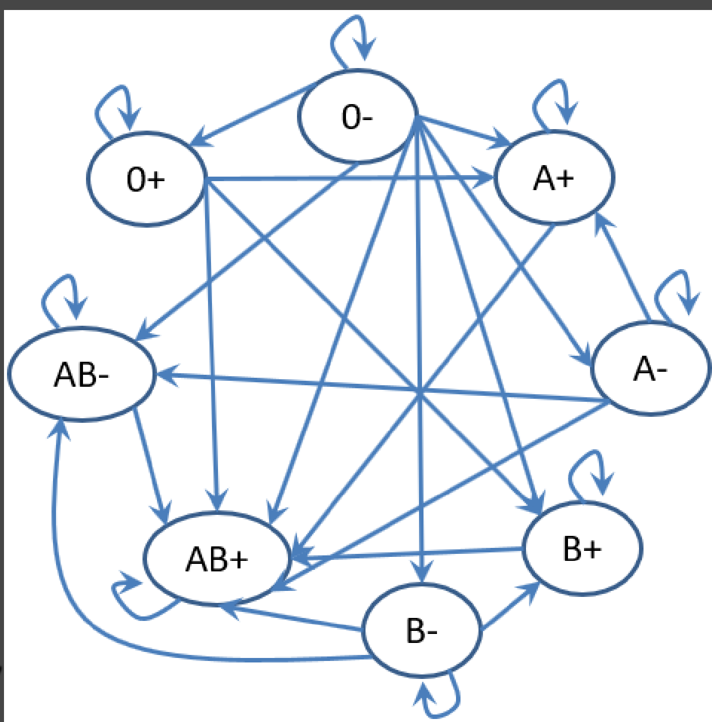


Σχεδιάστε όλα τα δυνατά δυαδικά δέντρα αναζήτησης που περιέχουν τις τιμές: 1, 2, 3.

Με βάση τον παρακάτω πίνακα συμβατότητας των ομάδων αίματος να φτιαχτεί ο ισοδύναμος κατευθυνόμενος γράφος.

Η ομάδα αίματός σας:

Μπορείτε να δεχθείτε από:	O+	O-	A+	A-	B+	B-	AB+	AB-
O+	♥		♥		♥		♥	
O-	♥	♥	♥	♥	♥	♥	♥	♥
A+			♥				♥	
A-			♥	♥			♥	♥
B+					♥		♥	
B-					♥	♥	♥	♥
AB+							♥	
AB-							♥	♥



Θέση μνήμης	Κόμβος	
	Τιμή	Δείκτης
		Κεφαλή: ...
100	X	...
200	...	400
300	I	...
400	...	300
500	O	...
600	Σ	...
700	...	200

Απλά συνδεδεμένη

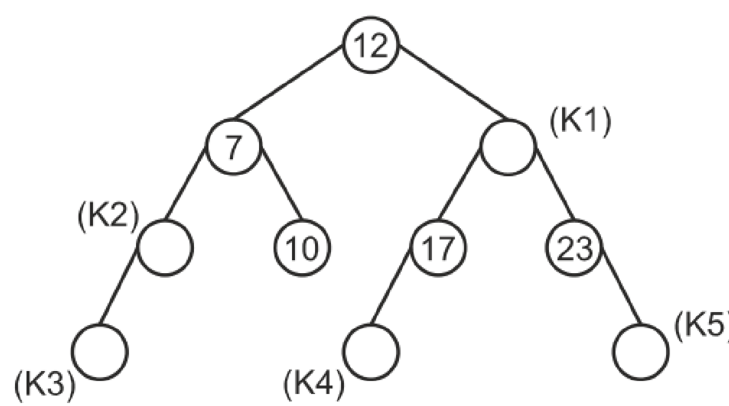
Συμπληρώστε τα κενά ώστε να προκύψει λίστα με τη λέξη "ΣΧΟΛΕΙΟ"

Κεφαλή	...
Ουρά	...

Θέση μνήμης	Κόμβος		
	Τιμή	Δείκτης Προηγούμενος	Δείκτης Επόμενος
100	Λ	...	...
200	...	700	...
300	I	...	...
400	...	...	0
500	E	...	...
600	O	...	...
700	Σ	...	...

Διπλά συνδεδεμένη

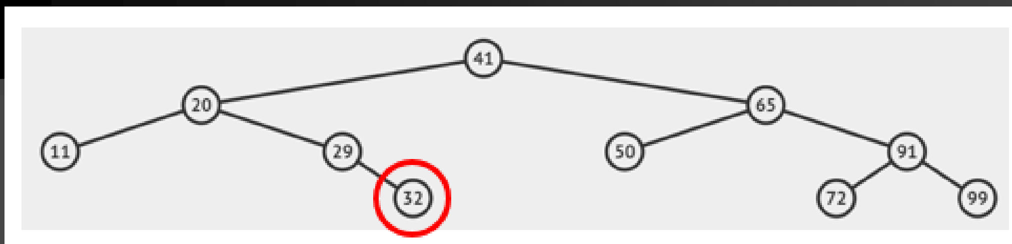
Για καθέναν από τους κόμβους να γράψετε στο τετράδιό σας τα Κ1, Κ2, Κ3, Κ4, Κ5 και δίπλα την κατάλληλη τιμή από τις τιμές: **4, 6, 15, 20, 34,** ώστε το δένδρο να είναι δυαδικό δένδρο αναζήτησης.

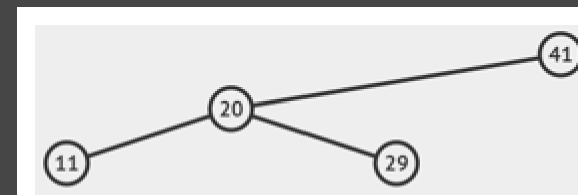


ΓΕΛ 2023

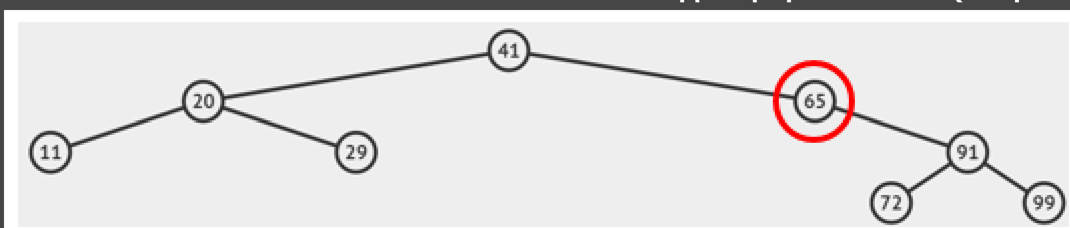
Αλγόριθμος διαγραφής κόμβου Κ από Δυαδικό Δέντρο Αναζήτησης:

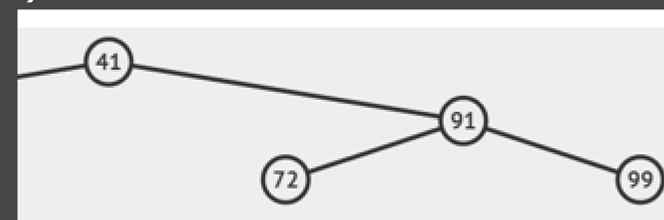
1. αν ο Κ είναι φύλλο, διέγραψε τον Κ
2. αν ο Κ έχει ένα παιδί τον κόμβο Π, παράκαμψε τον Κ "ανεβάζοντας" τον Π στη θέση του Κ
3. διαφορετικά, αντικατέστησε τον Κ με τον μικρότερο κόμβο του δεξιού του υποδέντρου (Μ) και διέγραψε το φύλλο Μ



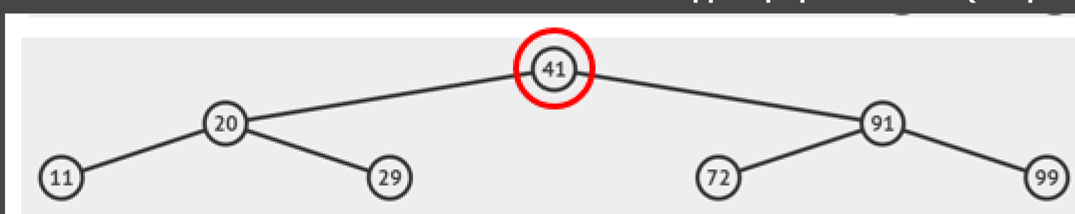


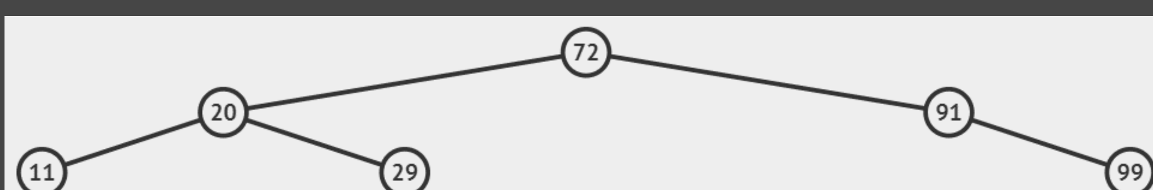
Διαγραφή του 32 (Περίπτωση 1)





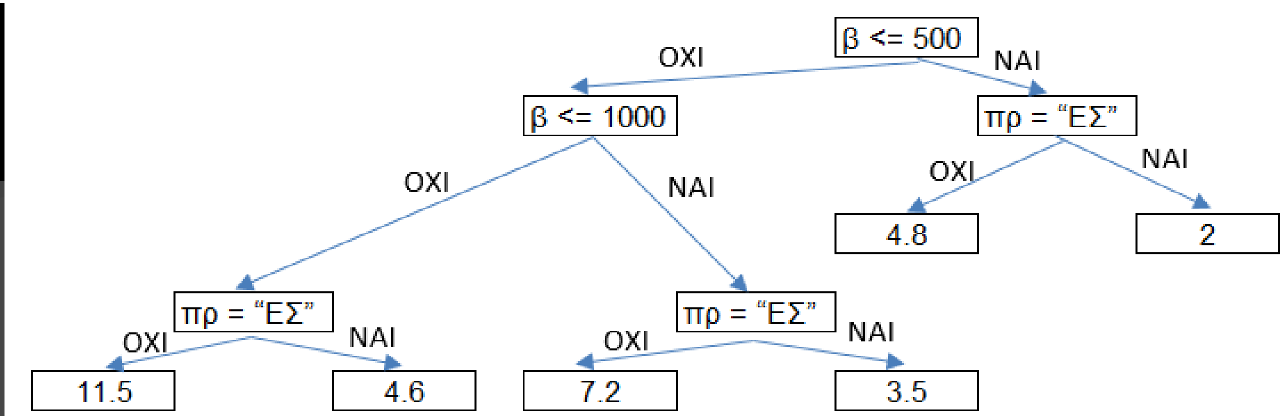
Διαγραφή του 65 (Περίπτωση 2)



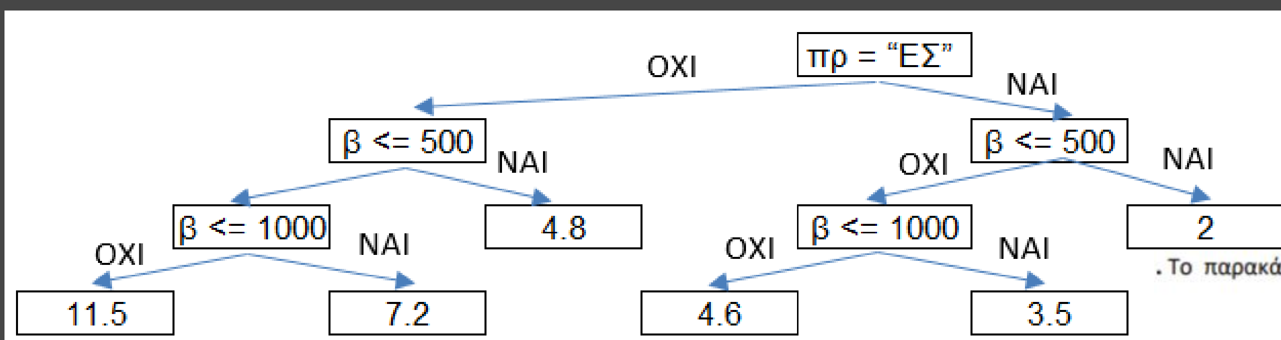


Διαγραφή του 41 (Περίπτωση 3)

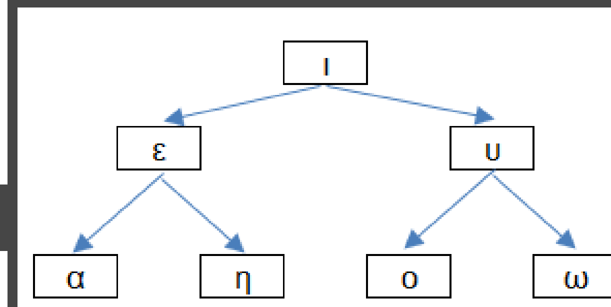
Βάρος επιστολής σε γραμμάρια	Χρέωση εσωτερικού σε €	Χρέωση εξωτερικού σε €
από 0 έως και 500	2,0	4,8
από 501 έως και 1000	3,5	7,2
από 1001 και άνω	4,6	11,5



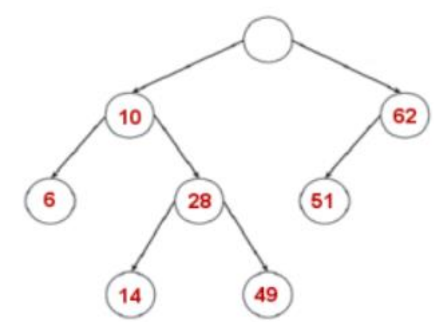
ή



Να φτιαχτεί το ισορροπημένο δυαδικό δέντρο αναζήτησης με τα φωνήεντα του ελληνικού αλφαβήτου.

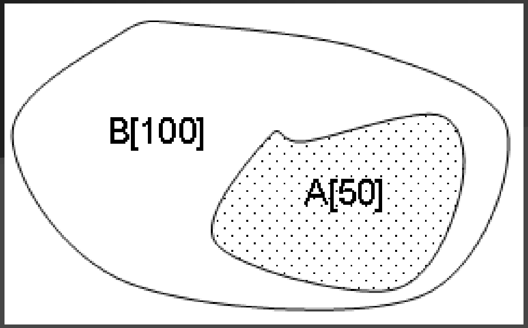


. Το παρακάτω δένδρο είναι δυαδικό δένδρο αναζήτησης.



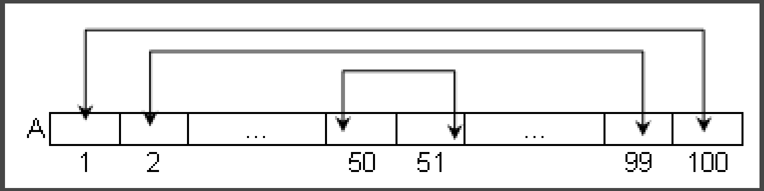
να συμπληρώσετε την τιμή της ρίζας, αν είναι γνωστό πως είναι ακέραια.

Σχέση υποσυνόλου π.χ. δίνονται οι A [50] και B[100]. Να ελεγχθεί εάν A ⊂ B




```
Π ← 0
για i από 1 μέχρι 50
  βρ ← Ψευδής
  j ← 1
  Όσο j <= 100 ΚΑΙ βρ = Ψευδής επανάλαβε
    Αν A[i] = B[j] τότε
      βρ ← Αληθής
    Αλλιώς
      j ← j + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν βρ = Αληθής τότε
    Π ← Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν Π = 50 τότε
  Γράψε "ναι"
Αλλιώς
  Γράψε "όχι"
ΤέλοςΑν
```

Έλεγχος για παλινδρομικότητα π.χ. δίνεται ο A [100]. Να ελεγχθεί αν είναι παλινδρομικός

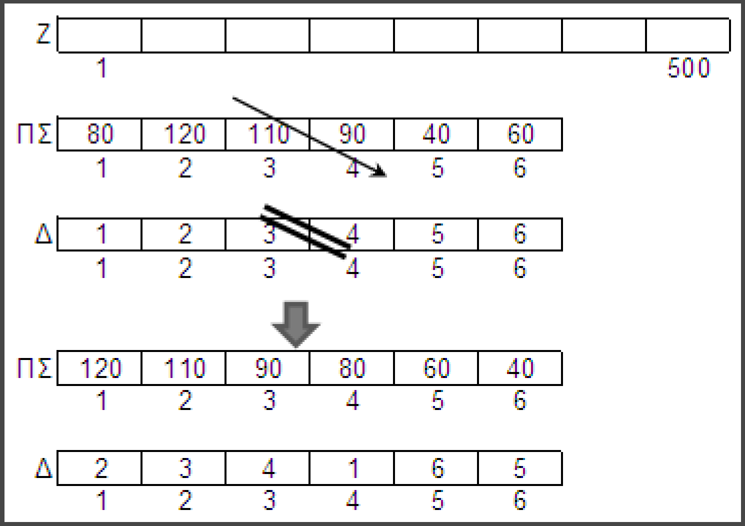



```
Π ← 0
για i από 1 μέχρι 50
  Αν A[i] = A[101-i] τότε
    Π ← Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν Π = 50 τότε
  Γράψε "ναι"
Αλλιώς
  Γράψε "όχι"
ΤέλοςΑν
```

Πίνακες: περισσότερες μεθοδολογίες

Παράλληλη ταξινόμηση πίνακα συχνοτήτων και πίνακα των δεικτών του π.χ. Δίνεται ο Z[500] με τις τυχαίες ενδείξεις των 500 ρίψεων ενός ζαριού (1-6). Να εμφανισθούν οι 6 ενδείξεις του ζαριού κατά φθίνουσα σειρά συχνότητας.

```
για i από 1 μέχρι 6
  ΠΣ[i] ← 0
  Δ[i] ← i
ΤέλοςΕπανάληψης
για i από 1 μέχρι 500
  ΠΣ[Z[i]] ← ΠΣ[Z[i]] + 1
  ! εναλλακτικά:
  ! x ← Z[i]
  ! ΠΣ[x] ← ΠΣ[x] + 1
ΤέλοςΕπανάληψης
για i από 2 μέχρι 6
  για j από 6 μέχρι i μεβήμα -1
    Αν ΠΣ[j-1] < ΠΣ[j] τότε
      tmp1 ← ΠΣ[j-1]
      ΠΣ[j-1] ← ΠΣ[j]
      ΠΣ[j] ← tmp1
      tmp2 ← Δ[j-1]
      Δ[j-1] ← Δ[j]
      Δ[j] ← tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 6
  Γράψε 'Ένδειξη: ', Δ[i], ' συχνότητα:', ΠΣ[i], 'φορές'
ΤέλοςΕπανάληψης
```



Πίνακες: περισσότερες μεθοδολογίες - **Αντιγραφές πινάκων**

1-Δ → 1-Δ ανάποδα π.χ. A[100] → B[100]

```
για i από 1 μέχρι 100
  B[i] ← A[101-i]
ΤέλοςΕπανάληψης
```

1-Δ → δύο 1-Δ π.χ. A[100] → B[50], Γ[50]

```
για i από 1 μέχρι 50
  B[i] ← A[i]
  Γ[i] ← A[i + 50]
ΤέλοςΕπανάληψης
```

2-Δ → 1-Δ π.χ. B[50, 100] → A[5000]

```
x ← 1
για i από 1 μέχρι 50
  για j από 1 μέχρι 100
    A[x] ← B[i, j]
    x ← x + 1
    ! ή A[(i-1)*100+j] ← B[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

1-Δ → 2-Δ π.χ. A[5000] → B[50, 100]

```
x ← 1
για i από 1 μέχρι 50
  για j από 1 μέχρι 100
    B[i, j] ← A[x]
    x ← x + 1
    ! ή B[i, j] ← A[(i-1)*100+j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

2-Δ → 2-Δ π.χ. A[30, 40] → B[10, 120]

A[30, 40] → X[1200]
X[1200] → B[10, 120]

ή

```
x ← 1 y ← 1
για i από 1 μέχρι 30
  για j από 1 μέχρι 40
    B[x, y] ← A[i, j]
    y ← y + 1
    Αν y = 121 τότε
      x ← x + 1
      y ← 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

B[80, 40] περιοχή 5x10 → A[5, 10] ξεκινώντας από το στοιχείο B[20, 40] (*crop*)

```
για i από 1 μέχρι 5
  για j από 1 μέχρι 10
    A[i, j] ← B[20+i-1, 40+j-1]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Πίνακες: περισσότερες μεθοδολογίες - **Αντιγραφές πινάκων**

2-Δ → δύο 2-Δ

π.χ. Γ[100, 160] → Α[100, 80], Β[100, 80]

```
για i από 1 μέχρι 100
  για j από 1 μέχρι 80
    Α[i, j] ← Γ[i, j]
    Β[i, j] ← Γ[i, j + 80]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

4 τεταρτημόρια 2-Δ → 4 2-Δ

π.χ. τεταρτημόρια Α[100, 100] →
Χ1[50, 50], Χ2[50, 50], Χ3[50, 50], Χ4[50, 50]

```
για i από 1 μέχρι 50
  για j από 1 μέχρι 50
    Χ1[i, j] ← Α[i, j]
    Χ2[i, j] ← Α[i, j+50]
    Χ3[i, j] ← Α[i+50, j]
    Χ4[i, j] ← Α[i+50, j+50]
ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

δύο 2-Δ → 2-Δ

π.χ. Α[100, 80], Β[100, 80] → Γ[100, 160]

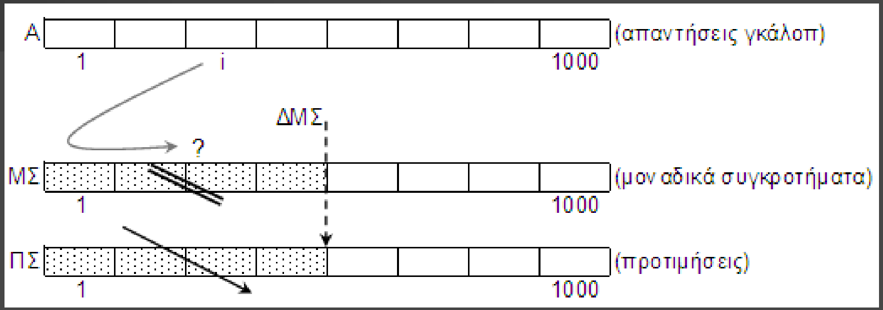
```
για i από 1 μέχρι 100
  για j από 1 μέχρι 80
    Γ[i, j] ← Α[i, j]
    Γ[i, j + 80] ← Β[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Α[50, 100] → Β[49, 99] αφαιρώντας τη γραμμή γρ και τη στήλη στ

```
για i από 1 μέχρι 50
  για j από 1 μέχρι 100
    Αν i < γρ ΚΑΙ j < στ τότε
      Β[i,j] ← Α[i,j]
    ΑλλιώςΑν i < γρ ΚΑΙ j > στ τότε
      Β[i,j-1] ← Α[i,j]
    ΑλλιώςΑν i > γρ ΚΑΙ j < στ τότε
      Β[i-1,j] ← Α[i,j]
    Αλλιώς
      Β[i-1,j-1] ← Α[i,j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Πίνακες: περισσότερες μεθοδολογίες

Συλλογή των διαφορετικών τιμών 1-Δ πίνακα π.χ. δίνεται ο A[1000] με τις απαντήσεις που δόθηκαν από 1000 άτομα σε ένα γκάλοπ, στην ερώτηση «ποιό είναι το αγαπημένο σας συγκρότημα;». Να εμφανισθούν τα <u>διαφορετικά</u> συγκροτήματα των απαντήσεων κατά ↓ σειρά προτιμήσεων.

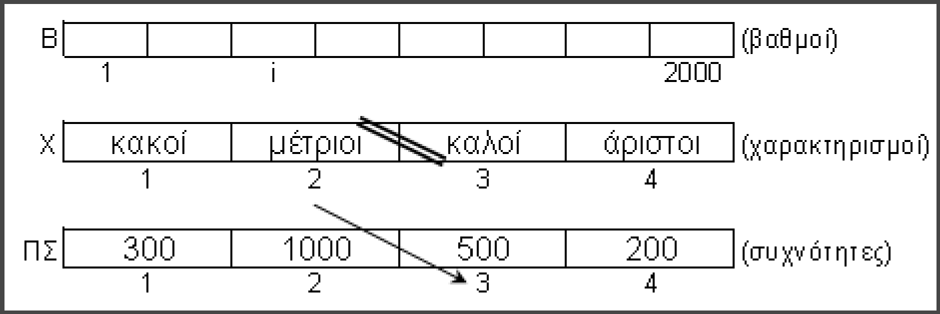



```
ΔΜΣ ← 0
για i από 1 μέχρι 1000
  ! αναζήτηση του A[i] στον ΜΣ από 1-ΔΜΣ
  βρ ← Ψευδής
  j ← 1
  Όσο j <= ΔΜΣ ΚΑΙ βρ = Ψευδής επανάλαβε
    Αν A[i] = ΜΣ[j] τότε
      βρ ← Αληθής
      θέση ← j
    Αλλιώς
      j ← j + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν βρ = Αληθής τότε
    ΠΣ[θέση] ← ΠΣ[θέση] + 1
  Αλλιώς
    ΔΜΣ ← ΔΜΣ + 1
    ΜΣ[ΔΜΣ] ← A[i]
    ΠΣ[ΔΜΣ] ← 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
! ↓ ταξινόμηση του ΠΣ και παράλληλα του ΜΣ
για i από 2 μέχρι ΔΜΣ
  για j από ΔΜΣ μέχρι i μεβήμα -1
    Αν ΠΣ[j-1] < ΠΣ[j] τότε
      tmp <-- ΠΣ[j-1]
      ΠΣ[j-1] <-- ΠΣ[j]
      ΠΣ[j] <-- tmp
      tmp2 <-- ΜΣ[j-1]
      ΜΣ[j-1] <-- ΜΣ[j]
      ΜΣ[j] <-- tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι ΔΜΣ
  Γράψε 'Συγκρότημα: ', ΜΣ[i], ' προτιμήσεις:', ΠΣ[i]
ΤέλοςΕπανάληψης
```

Πίνακες: περισσότερες μεθοδολογίες

Πίνακας συχνοτήτων διαστημάτων τιμών π.χ. δίνεται ο B[2000] με τους βαθμούς 2000 μαθητών σε ένα μάθημα ([0, 100]). Δίνεται ο πίνακας χαρακτήρων X[4] με τους χαρακτηρισμούς (X[1] = "κακοί" [0, 50), X[2] = "μέτριοι" [50, 70), X[3] = "καλοί" [70, 90), X[4] = "άριστοι" [90, 100]), Να εμφανισθούν οι παραπάνω χαρακτηρισμοί κατά φθίνουσα σειρά συχνοτήτων.



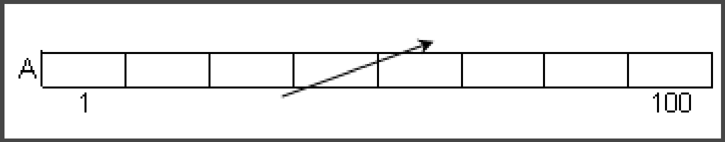

```
για i από 1 μέχρι 4
  ΠΣ[i] ← 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 2000
  Αν (B[i] < 50) τότε
    ΠΣ[1] ← ΠΣ[1] + 1
  ΑλλιώςΑν (B[i] < 70) τότε
    ΠΣ[2] ← ΠΣ[2] + 1
  ΑλλιώςΑν (B[i] < 90) τότε
    ΠΣ[3] ← ΠΣ[3] + 1
  Αλλιώς
    ΠΣ[4] ← ΠΣ[4] + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
για i από 2 μέχρι 4
  για j από 4 μέχρι i μεβήμα -1
    Αν ΠΣ[j-1] < ΠΣ[j] τότε
      tmp <-- ΠΣ[j-1]
      ΠΣ[j-1] <-- ΠΣ[j]
      ΠΣ[j] <-- tmp
      tmp2 <-- X[j-1]
      X[j-1] <-- X[j]
      X[j] <-- tmp2ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 4
  Γράψε X[i], ' συχνότητα:', ΠΣ[i]
ΤέλοςΕπανάληψης
```

Πίνακες: περισσότερες μεθοδολογίες

"Έξυπνη" φυσαλίδα: αντιλαμβάνεται πότε ο πίνακας είναι ταξινομημένος και σταματάει τους ελέγχους. Εάν δηλ. σε κάποιο «πέρασμα» του j δεν γίνει καμία αντιμετάθεση αυτό σημαίνει ότι ο πίνακας είναι ταξινομημένος.

```
ταξινομημένος ← Ψευδής
i ← 2
Όσο i <= 100 ΚΑΙ  ταξινομημένος = Ψευδής επανάλαβε
  ΠΑ ← 0 ! πλήθος αντιμεταθέσεων
  για j από 100 μέχρι i μεβήμα -1
    Αν A[j-1] > A[j] τότε
      tmp <-- A[j-1]
      A[j-1] <-- A[j]
      A[j] <-- tmp
      ΠΑ ← ΠΑ + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν ΠΑ = 0 τότε
    ταξινομημένος ← Αληθής
  ΤέλοςΑν
  i ← i + 1
ΤέλοςΕπανάληψης
```



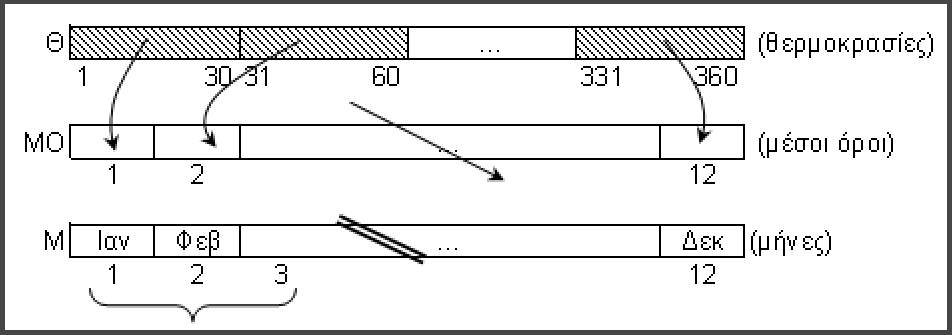
Πίνακες: περισσότερες μεθοδολογίες

```
για i από 1 μέχρι 12
  ΜΟ[i] ← 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 360
  μήνας ← i div 30
  Αν i mod 30 <> 0 τότε
    μήνας ← μήνας + 1
 ΤέλοςΑν
  ΜΟ[μήνας] ← ΜΟ[μήνας] + Θ[i]
ΤέλοςΕπανάληψης
για i από 1 μέχρι 12
  ΜΟ[i] ← ΜΟ[i] / 30
ΤέλοςΕπανάληψης
για i από 2 μέχρι 12
  για j από 12 μέχρι i μεβήμα -1
    Αν ΜΟ[j-1] < ΜΟ[j] τότε
      tmp <-- ΜΟ[j-1]
      ΜΟ[j-1] <-- ΜΟ[j]
      ΜΟ[j] <-- tmp
      tmp2 <-- Μ[j-1]
      Μ[j-1] <-- Μ[j]
      Μ[j] <-- tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 3
  Γράψε Μ[i]
ΤέλοςΕπανάληψης
```

ή

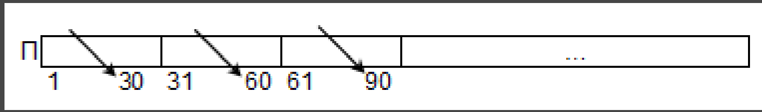
```
για μήνας από 1 μέχρι 12
  για j από 30*μήνας-29 μέχρι 30*μήνας
    ΜΟ[μήνας] ← ΜΟ[μήνας] + Θ[j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Ομαδοποίηση πίνακα: π.χ. δίνεται ο Θ[360] με τις θερμοκρασίες μιας πόλης στις 12:00 το μεσημέρι για κάθε ημέρα ενός έτους (1 μήνας = 30 ημέρες). Να δημιουργηθεί ο ΜΟ[12] με τις μέσες θερμοκρασίες του κάθε μήνα και δεδομένου του Μ[12] με τα ονόματα των 12 μηνών να εμφανισθούν οι 3 πιο θερμοί μήνες.



Πολλαπλή ταξινόμηση υποπεριοχών 1-Δ πίνακα: π.χ. δίνεται ο Π[360] με τις πωλήσεις μιας εταιρείας για κάθε ημέρα ενός έτους (1 μήνας = 30 ημέρες). Πρόγραμμα που δεδομένου του Μ[12] με τα ονόματα των 12 μηνών, εμφανίζει τον κάθε μήνα ακολουθούμενο από τις 5 μεγαλύτερες πωλήσεις του.

```
για μήνας από 1 μέχρι 12
  ημ1 ← (μήνας – 1) * 30 + 1
  ημ2 ← μήνας * 30
  για i από ημ1 + 1 μέχρι ημ2
    για j από ημ2 μέχρι i μεβήμα -1
      Αν Π[j-1] < Π[j] τότε
        tmp <-- Π[j-1]
        Π[j-1] <-- Π[j]
        Π[j] <-- tmp
      ΤέλοςΑν
    ΤέλοςΕπανάληψης
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για μήνας από 1 μέχρι 12
  ημ1 ← (μήνας – 1) * 30 + 1
  Γράψε 'Για το μήνα', Μ[μήνας]
  για i από ημ1 μέχρι ημ1 + 4
    Γράψε Π[i]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

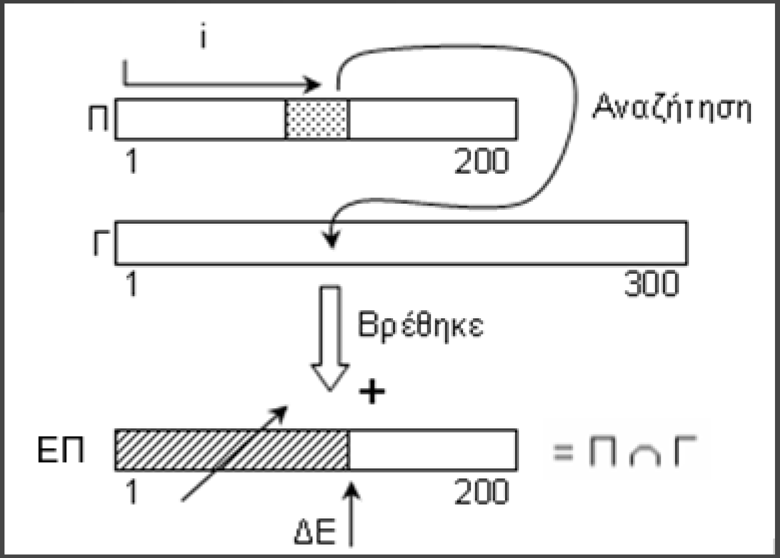


Αναζήτηση 2 τιμών. Ποια βρέθηκε πρώτη; Πρόγραμμα που δεδομένου του ακέραιου A[100], και δύο τιμών x1, x2 (διαφορετικές μεταξύ τους) και τις αναζητά στον A. Εμφανίζει ποια εντόπισε πρώτη και που ή μήνυμα αν δεν εντόπισε καμία.

```
f1 <-- Ψευδής
f2 <-- Ψευδής
i <-- 1
Όσο i <= 100 ΚΑΙ f1 = Ψευδής ΚΑΙ
f2 = Ψευδής επανάλαβε
  Αν x1 = A[i] τότε
    f1 <-- Αληθής
    pos1 <-- i
  ΑλλιώςΑν x2 = A[i] τότε
    f2 <-- Αληθής
    pos2 <-- i
  ΤέλοςΑν
  i <-- i + 1
ΤέλοςΕπανάληψης
Αν f1 = Αληθής τότε
  Γράψε 'Βρέθηκε η 1η:', pos1
ΑλλιώςΑν f2 = Αληθής τότε
  Γράψε 'Βρέθηκε η 2η:', pos2
Αλλιώς
  Γράψε 'Δε βρέθηκε καμία'
ΤέλοςΑν
```

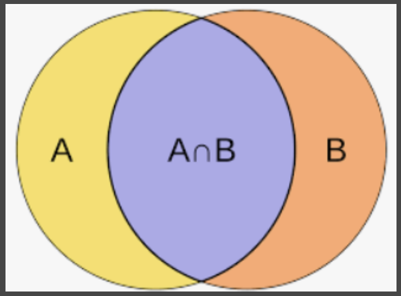
Πίνακες: περισσότερες μεθοδολογίες

Τομή πινάκων: π.χ. δίνονται ο Π[200] και ο Γ[300] με τα ονόματα των επιτυχόντων στα προφορικά και στα γραπτά ενός διαγωνισμού του ΑΣΕΠ. Να εμφανισθούν αλφαβητικά οι επιτυχόντες και στις 2 δοκιμασίες (Π ∩ Γ).




```
ΔΕ ← 0
για i από 1 μέχρι 200
  ! Αναζήτηση του Π[i] στον Γ[300]
  j ← 1
  βρ ← Ψευδής
  Όσο j <= 300 ΚΑΙ βρ = Ψευδής επανάλαβε
    Αν Γ[j] = Π[i] τότε
      βρ ← Αληθής
    Αλλιώς
      j ← j + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν βρ = Αληθής τότε
    ΔΕ ← ΔΕ + 1
    ΕΠ[ΔΕ] ← Π[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

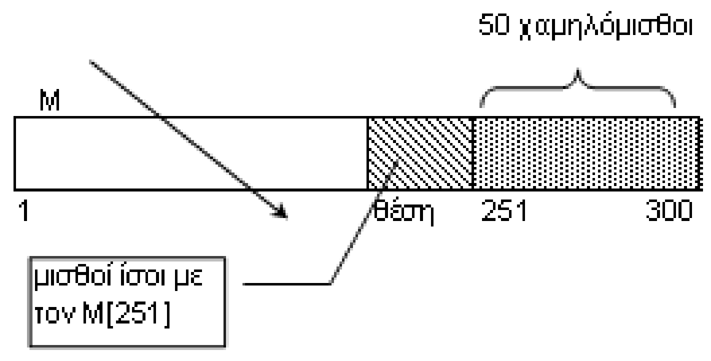
```
! Ταξινόμηση ↑ του ΕΠ από το 1ο ως το ΔΕ κελί
για i από 2 μέχρι ΔΕ
  για j από ΔΕ μέχρι i μεβήμα -1
    Αν ΕΠ[j-1] > ΕΠ[j] τότε
      tmp <-- ΕΠ[j-1]
      ΕΠ[j-1] <-- ΕΠ[j]
      ΕΠ[j] <-- tmp
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι ΔΕ
  Γράψε ΕΠ[i]
ΤέλοςΕπανάληψης
```



Πίνακες: περισσότερες μεθοδολογίες

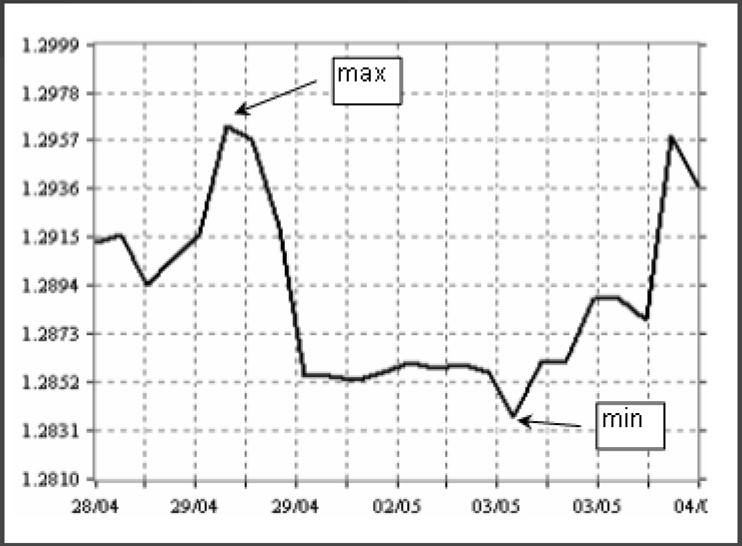
```
! Ταξινόμηση ↓ του M[300] και παράλληλα του O[300]
για i από 2 μέχρι 300
  για j από 300 μέχρι i μεβήμα -1
    Αν M[j-1] < M[j] τότε
      tmp <-- M[j-1]
      M[j-1] <-- M[j]
      M[j] <-- tmp
      tmp2 <-- O[j-1]
      O[j-1] <-- O[j]
      O[j] <-- tmp2
ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
! εύρεση του 1ου μισθού στην 50η θέση
θέση ← 250
Όσο θέση >= 1 ΚΑΙ M[θέση] = M[251] επανάλαβε
  θέση ← θέση - 1
ΤέλοςΕπανάληψης
θέση ← θέση + 1
για i από θέση μέχρι 300
  M[i] ← M[i]  + 5/100 * M[i]
  Γράψε O[i], M[i]
ΤέλοςΕπανάληψης
Γράψε 'Σύνολο: ', 300 – θέση + 1, 'υπάλληλοι'
```

Πρόβλεψη ισοβαθμίας: π.χ. Δίνονται ο O[300] και ο M[300] με τα ονόματα και τους μισθούς 300 υπαλλήλων μιας εταιρείας. Η εταιρεία αποφάσισε να δώσει 5% αύξηση στους 50 πιο χαμηλόμισθους. Πρόγραμμα που ενημερώνει τον πίνακα των μισθών (προβλέποντας την περίπτωση ισοβαθμίας στην 50η μισθολογική θέση) και εμφανίζει ποιοι και πόσοι υπάλληλοι παίρνουν αύξηση, μαζί με το νέο μισθό τους.



Πίνακες: περισσότερες μεθοδολογίες

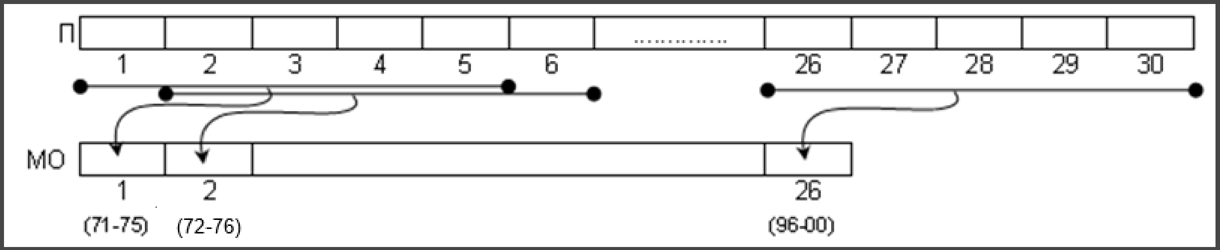
Εξελικτική πορεία τιμής: π.χ. Δίνεται οι T[360] με τις ημερήσιες τιμές του ενός βαρελιού πετρελαίου για 1 έτος (1 μήνας = 30 ημέρες). Πρόγραμμα που εμφανίζει τη μεγαλύτερη % άνοδο και πτώση της τιμής μεταξύ 2 διαδοχικών ημερών.




```
για i από 1 μέχρι 359
  M[i] ← (T[i+1] – T[i]) / T[i] * 100
ΤέλοςΕπανάληψης
ΠΘΜ ← 0 ! πλήθος θετικών μεταβολών
ΠΑΜ ← 0 ! πλήθος αρνητικών μεταβολών
για i από 1 μέχρι 359
  Αν M[i] > 0 τότε
    ΠΘΜ ← ΠΘΜ + 1
    Αν ΠΘΜ = 1 τότε
      max ← M[i]
    ΑλλιώςΑν M[i] > max τότε
      max ← M[i]
    ΤέλοςΑν
  ΑλλιώςΑν M[i] < 0 τότε
    ΠΑΜ ← ΠΑΜ + 1
    Αν ΠΑΜ = 1 τότε
      min ← M[i]
    ΑλλιώςΑν M[i] < min τότε
      min ← M[i]
    ΤέλοςΑν
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν ΠΘΜ <> 0 τότε
  Γράψε 'Μέγιστη άνοδος: ', max, '%'
Αλλιώς
  Γράψε 'Καμία άνοδος'
ΤέλοςΑν
Αν ΠΑΜ <> 0 τότε
  Γράψε 'Μέγιστη πτώση: ', min, '%'
Αλλιώς
  Γράψε 'Καμία πτώση'
ΤέλοςΑν
```

Πίνακες: περισσότερες μεθοδολογίες

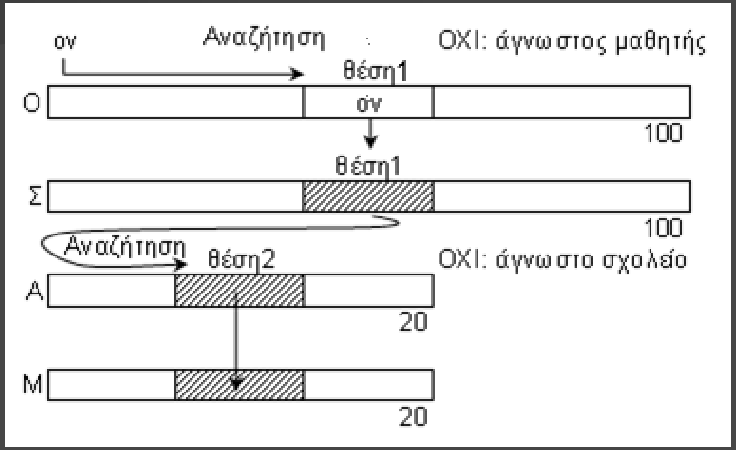
```
! γκρουπάρισμα των 5ετιών
για i από 1 μέχρι 26
  s ← 0
  για j από i μέχρι i + 4
    s ← s + Π[j]
  τέλοςΕπανάληψης
  ΜΟ[i] ← s / 5
τέλοςΕπανάληψης
! εύρεση του max του ΜΟ[26]
max ← ΜΟ[1]
για i από 2 μέχρι 26
  Αν ΜΟ[i] > max τότε
    max ← ΜΟ[i]
 ΤέλοςΑν
τέλοςΕπανάληψης
! εύρεση των αποδοτικότερων 5ετιών
για i από 1 μέχρι 26
  Αν ΜΟ[i] = max τότε
    Γράψε ‛Πενταετία: ‛
    Γράψε 1970 + i, ‛ – ‛, 1970 + i + 4
  ΤέλοςΑν
τέλοςΕπανάληψης
```



Διαδοχικές ομάδες κελιών: π.χ. Δίνεται ο Π[30] με τις ετήσιες πωλήσεις μιας Εταιρείας για μία περίοδο 30 ετών (1971-2000). Πρόγραμμα που εμφανίζει την 5ετία (ή τις 5ετίες) πέντε συνεχόμενων ετών, με το μεγαλύτερο μέσο όρο πωλήσεων.

Πίνακες: περισσότερες μεθοδολογίες

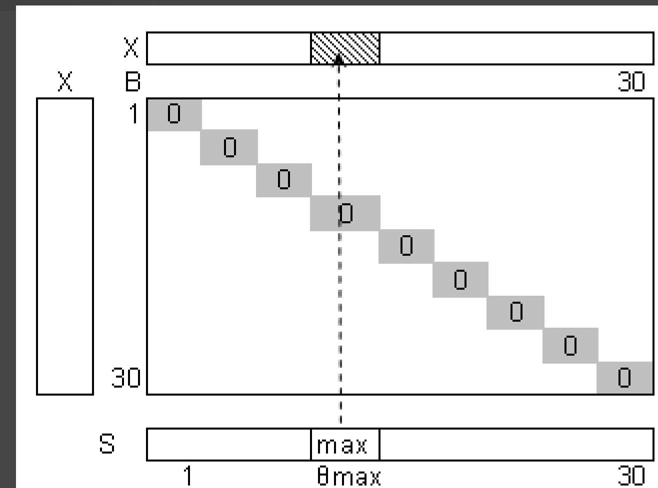
Αναζήτηση 2 επιπέδων: π.χ. Δίνονται οι Ο[100] και Σ[100] με τα ονόματα και τα σχολεία 100 μαθητών. Οι μαθητές φοιτούν σε 20 διαφορετικά σχολεία που έχουν καταγραφεί στον Α[20], καθώς και τα email τους στον Μ[20]. Πρόγραμμα που διαβάζει το όνομα ενός μαθητή και εμφανίζει το email του σχολείου του.




```
Διάβασε ον
! αναζήτηση μαθητή ον στον Ο[100]
βρ ← Ψευδής
i ← 1
Όσο i <= 100 ΚΑΙ βρ = Ψευδής επανάλαβε
  Αν (ον = Ο[i]) τότε
    βρ ← Αληθής
    θέση1 ← i
  Αλλιώς
    i ← i + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν βρ = Αληθής τότε
  ! αναζήτηση σχολείου Σ[θέση1] στον Α[20]
  βρ ← Ψευδής
  i ← 1
  Όσο i <= 20 ΚΑΙ βρ = Ψευδής επανάλαβε
    Αν Σ[θέση1] = Α[i] τότε
      βρ ← Αληθής
      θέση2 ← i
    Αλλιώς
      i ← i + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν βρ = Αληθής τότε
    Γράψε Μ[θέση2]
  Αλλιώς
    Γράψε 'Δεν βρέθηκε το σχολείο'
  ΤέλοςΑν
Αλλιώς
  Γράψε 'Δεν βρέθηκε ο μαθητής'
ΤέλοςΑν
```

Πίνακες: περισσότερες μεθοδολογίες

Πίνακας αξιολόγησης: π.χ. Σε έναν διαγωνισμό τραγουδιού συμμετέχουν 30 χώρες. Τα ονόματά τους δίνονται στον X[30]. Κάθε χώρα βαθμολογεί 12 άλλες χώρες με τους βαθμούς 1, 2, ..., 12 και δεν αυτοαξιολογείται. Πρόγραμμα που αρχικά μηδενίζει τον B[30, 30] και στη συνέχεις δέχεται σε αυτόν τις βαθμολογίες των χωρών, έτσι ώστε στο B[i, j] να βρίσκεται ο βαθμός που έδωσε η χώρα X[i] στη χώρα X[j] (ή το 0). Εμφανίζει τη νικήτρια χώρα (χωρίς περίπτωση ισοβαθμίας)

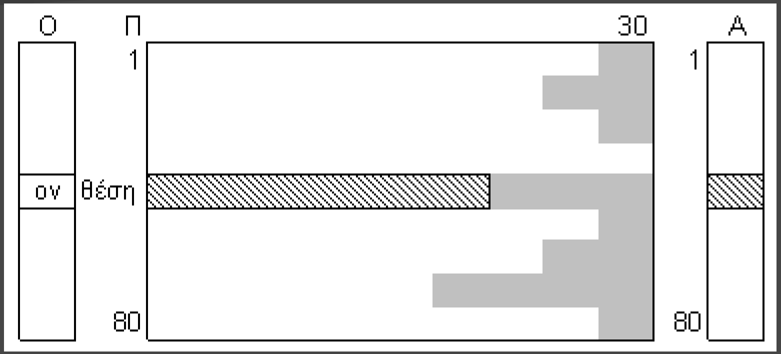



```
για i από 1 μέχρι 30
  για j από 1 μέχρι 30
    B[i, j] ← 0
  τέλοςΕπανάληψης
τέλοςΕπανάληψης
για i από 1 μέχρι 30
  Γράψε 'Βαθμολογεί η χώρα: ', X[i]
  για j από 1 μέχρι 12
    Γράψε 'Βαθμός ', j, ' στη χώρα:'
      ΑρχήΕπανάληψης
        Γράψε 'Δώσε αριθμό χώρας:'
        Διάβασε κ
      ΜέχριςΌτου κ >= 1 ΚΑΙ κ <= 30 ΚΑΙ κ <> i  ΚΑΙ B[i, κ] = 0
      B[i, κ] ← j
  τέλοςΕπανάληψης
τέλοςΕπανάληψης
για j από 1 μέχρι 30
  S[j] ← 0
  για i από 1 μέχρι 30
    S[j] ← S[j] + B[i.j]
  τέλοςΕπανάληψης
τέλοςΕπανάληψης
```


```
max ← S[1]
θmax ← 1
για i από 2 μέχρι 30
  Αν S[i] > max τότε
    max ← S[i]
    θmax ← i
  ΤέλοςΑν
τέλοςΕπανάληψης
Γράψε X[θmax]
```

Πίνακες: περισσότερες μεθοδολογίες

“Ημιγεμάτος” 2-Δ πίνακας (με τιμή φρουρό): π.χ. Εισαγωγή για 80 πωλητές: όνομα και οι ημερήσιες πωλήσεις που έκαναν για έναν μήνα (το πολύ 30), μέχρι να εξαντληθούν οι ημέρες ή να δοθεί τιμή ≤ 0 (τιμή φρουρός). Εισαγωγή του ονόματος ενός πωλητή του και εμφάνιση της μέσης μηνιαίας πώλησής του.



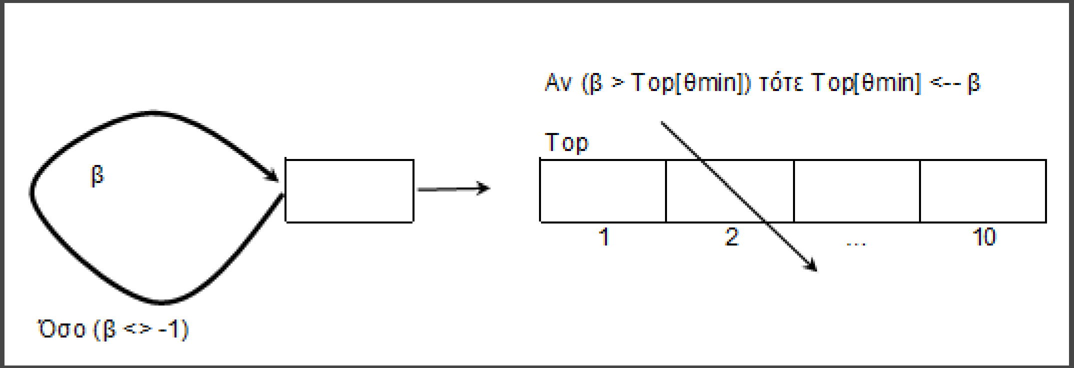

```
για i από 1 μέχρι 80
   Διάβασε Ο[i]
   Διάβασε πωλ
   j ← 0
   Όσο πωλ > 0 ΚΑΙ j < 30 επανάλαβε
      j ← j + 1
      Π[i, j] ← πωλ
     Διάβασε πωλ
   τέλοςΕπανάληψης
   Α[i] ← j ! πλήθος πωλήσεων ανά πωλητή
τέλοςΕπανάληψης
Διάβασε ον
βρ ← Ψευδής
i ← 1
Όσο i <= 80 ΚΑΙ βρ = Ψευδής επανάλαβε
  Αν ον = Ο[i] τότε
    βρ ← Αληθής
    θέση ← i
  Αλλιώς
    i ← i + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
Αν βρ = Αληθής τότε
  s ← 0
   για j από 1 μέχρι Α[θέση]
      s ← s + Π[θέση, j]
   τέλοςΕπανάληψης
   Αν Α[θέση] <> 0 τότε
      Γράψε s / Α[θέση], ‘€’
   Αλλιώς
      Γράψε ‘καμία πώληση’
   ΤέλοςΑν
Αλλιώς
  Γράψε ‘δε βρέθηκε’
ΤέλοςΑν
```

Πίνακες: περισσότερες μεθοδολογίες

```
για i από 1 μέχρι 10
 Top[i] ← -1
τέλοςΕπανάληψης
Διάβασε β
Όσο β <> -1 επανάλαβε
  ! εύρεση του θmin του Top
  θmin ← 1 min ← Top[1]
  για j από 2 μέχρι 10
    Αν Top[i] < min τότε
      min ← Top[i]
      θmin ← i
    ΤέλοςΑν
  τέλοςΕπανάληψης
  Αν β > Top[θmin] τότε
    Top[θmin] <-- β
  ΤέλοςΑν
  Διάβασε β
τέλοςΕπανάληψης
! Φθίνουσα ταξινόμηση του Top[10] …
για i από 1 μέχρι 10
  Αν Top[i] <> -1 τότε
    Γράψε Top[i]
  ΤέλοςΑν
τέλοςΕπανάληψης
```

Top min/max στατιστικά από σύνολο αγνώστου πλήθους στοιχείων: π.χ. εισαγωγή βαθμών μέχρι να δοθεί το (-1). Ποιοί οι 10 μεγαλύτεροι βαθμοί;

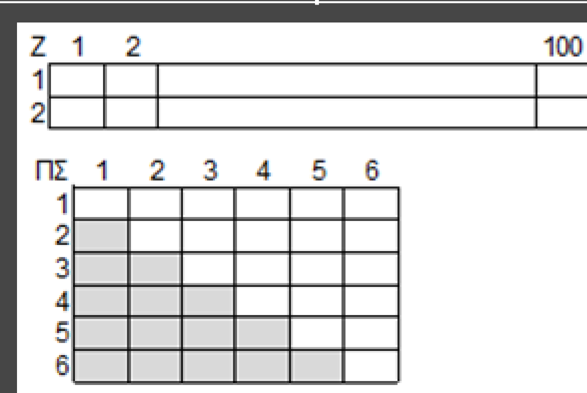


Πίνακες: περισσότερες μεθοδολογίες

```
! μηδενισμός του ΠΣ
για i από 1 μέχρι 6
  για j από i μέχρι 6
    ΠΣ[i, j] ← 0
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
! γέμισμα του ΠΣ
για i από 1 μέχρι 100
  z1 ← Z[1, i]
  z2 ← Z[2, i]
  Αν z1 <= z2 τότε
    ΠΣ[z1, z2] ← ΠΣ[z1, z2] + 1
  Αλλιώς
    ΠΣ[z2, z1] ← ΠΣ[z2, z1] + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
! max του ΠΣ
max ← ΠΣ[1,1]
για i από 1 μέχρι 6
  για j από i μέχρι 6
    Αν (ΠΣ[i, j] > max) τότε
      max ← ΠΣ[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 6
  για j από i μέχρι 6
    Αν ΠΣ[i, j] = max τότε
      Γράψε 'Ζαριά: ', i, j
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

2-Δ πίνακας συχνοτήτων: Δίνεται ο Z[2, 100] με τις ενδείξεις των 100 ρίψεων 2 ζαριών. Ποια(ες) ζαριά ήταν η συχνότερη; (οι ζαριές όπως π.χ. 1-2 και 2-1 να θεωρούνται οι ίδιες)

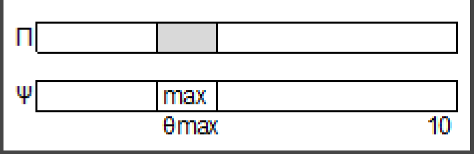


Πίνακες: περισσότερες μεθοδολογίες

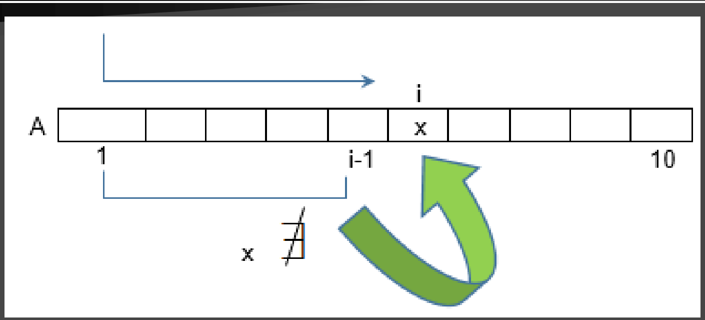

```
για i από 1 μέχρι 10
  Ψ[i] <-- 0
τέλοςΕπανάληψης
pa ← 0
ΑρχήΕπανάληψης
  Διάβασε πρ
  pa ← pa + 1
  ! αναζήτηση του πρ στον Π[10]
  βρ <-- Ψευδής
  i <-- 1
  Όσο i <= 10 ΚΑΙ βρ = Ψευδής
επανάλαβε
    Αν (Π[i] = πρ) τότε
      βρ <-- Αληθής
      θέση <-- i
    ΤέλοςΑν
    i <-- i + 1
  τέλοςΕπανάληψης
  Αν βρ τότε
    Ψ[θέση] <-- Ψ[θέση] + 1
  ΤέλοςΑν
```

```
! καταμέτρηση των προορισμών που ψηφίστηκαν
  p <-- 0
  για i από 1 μέχρι 10
    Αν Ψ[i] <> 0 τότε
      p <-- p + 1
    ΤέλοςΑν
  τέλοςΕπανάληψης
ΜέχριςΌτου pa>=100 ΚΑΙ p = 10
! max, θmax στον Ψ[10]
max <-- Ψ[1]
θmax <-- 1
για i από 2 μέχρι 10
  Αν Ψ[i] < max τότε
    max <-- Ψ[i]
    θmax <-- i
  ΤέλοςΑν
τέλοςΕπανάληψης
Γράψε Π[θmax]
```

"Επίσκεψη" όλων των στοιχείων πίνακα από μία τουλάχιστον φορά:
Δίνεται ο Π[10] με τα ονόματα 10 δημοφιλών τουριστικών προορισμών. Να γίνει γκάλοπ με την ερώτηση "ποιος είναι ο αγαπημένος σας τουριστικός προορισμός;". Το γκάλοπ να ζητάει επαναληπτικά απαντήσεις μέχρι να δοθούν 100 τουλάχιστον απαντήσεις και να ψηφιστούν και οι 10 προορισμοί από μία τουλάχιστον φορά. Ποιος προορισμός εκλέχθηκε ως ο δημοφιλέστερος; (θεωρείστε ότι είναι μόνο ένας)

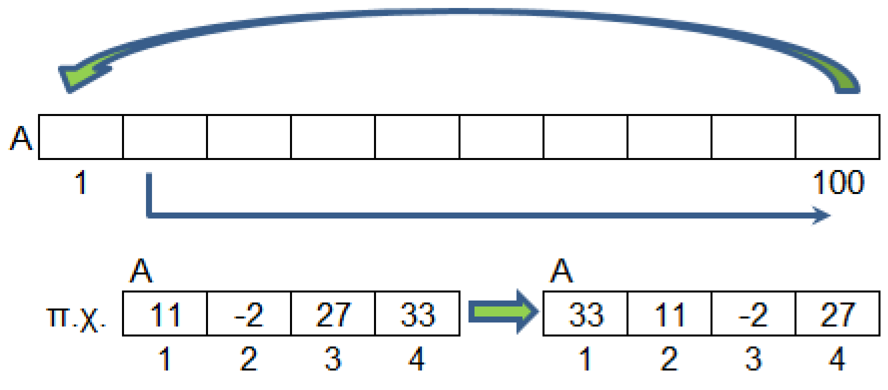


Εισαγωγή διαφορετικών τιμών σε πίνακα:
Πρόγραμμα που διαβάζει στον ακέραιο Α[10] τιμές που είναι διαφορετικές, με έλεγχο εγκυρότητας ως εξής: να διαβάζει επαναληπτικά την κάθε τιμή και να την αποδέχεται εάν είναι διαφορετική από όλες τις άλλες που έχουν εισαχθεί μέχρι τότε.




```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ 'Δώσε την ', i, 'η τιμή'
    ΔΙΑΒΑΣΕ  x
    j ← 1
    f ← ΨΕΥΔΗΣ
    ΟΣΟ j <= i-1 ΚΑΙ f = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
      ΑΝ x = A[j] ΤΟΤΕ
        f ← ΑΛΗΘΗΣ
      ΑΛΛΙΩΣ
        j ← j + 1
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΜΕΧΡΙΣ_ΟΤΟΥ f = ΨΕΥΔΗΣ
  A[i] ← x
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Κυκλική μετακίνηση τιμών σε 1-Δ πίνακα:
Πρόγραμμα που σε πίνακα Α[100] μετακινεί τις τιμές του κυκλικά κατά μία θέση προς τα δεξιά: το Α[1] στο Α[2], το Α[2] στο Α[3], … το Α[100] στο Α[1]

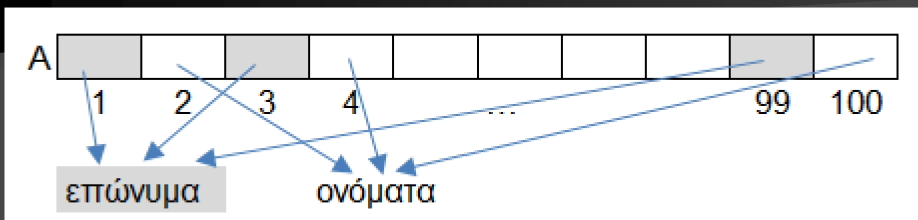



```
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
  A[i] ← A[i-1]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
A[1] ← A[100]
! Είναι σωστό;
```

```
last ← A[100]
ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 2 ΜΕΒΗΜΑ -1
  A[i] ← A[i-1]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
A[1] ← last
```

Ταξινόμηση ανά 2

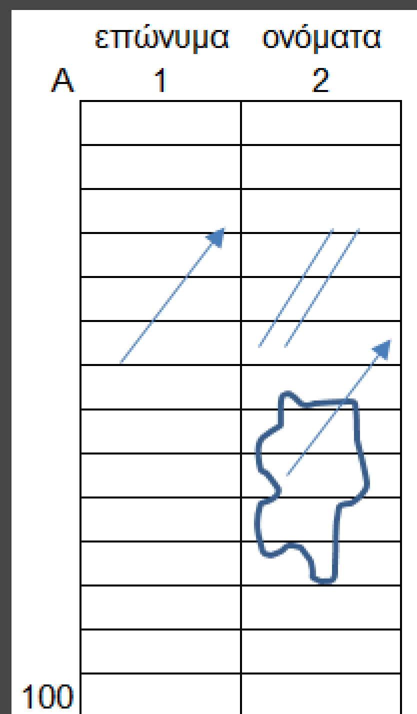
Δίνεται ο A[100] με τα επώνυμα (στοιχεία 1,3,...,99) και τα ονόματα (στοιχεία 2,4,...100) 50 μαθητών. Να ταξινομηθούν αλφαβητικά ως προς τα επώνυμα και οι συνώνυμοι στα επώνυμα, αλφαβητικά ως προς τα ονόματα (χωρίς τη χρήση άλλων πινάκων)




```
για i από 3 μέχρι 99 μεβήμα 2
  για j από 99 μέχρι i μεβήμα -2
    Αν A[j-2] > A[j] τότε
      tmp <-- A[j]
      A[j] <-- A[j-2]
      A[j-2] <-- tmp
      tmp <-- A[j+1]
      A[j+1] <-- A[j-1]
      A[j-1] <-- tmp
    ΑλλιώςΑν A[j-2] = A[j] ΚΑΙ
A[j-1] > A[j+1] τότε
      tmp <-- A[j+1]
      A[j+1] <-- A[j-1]
      A[j-1] <-- tmp
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Ταξινόμηση ανά 2

Δίνεται ο A[100, 2] με τα επώνυμα (1η στήλη) και τα ονόματα (2η στήλη) 100 μαθητών. Να ταξινομηθούν αλφαβητικά ως προς τα επώνυμα και οι συνώνυμοι στα επώνυμα, αλφαβητικά ως προς τα ονόματα (χωρίς τη χρήση άλλων πινάκων)

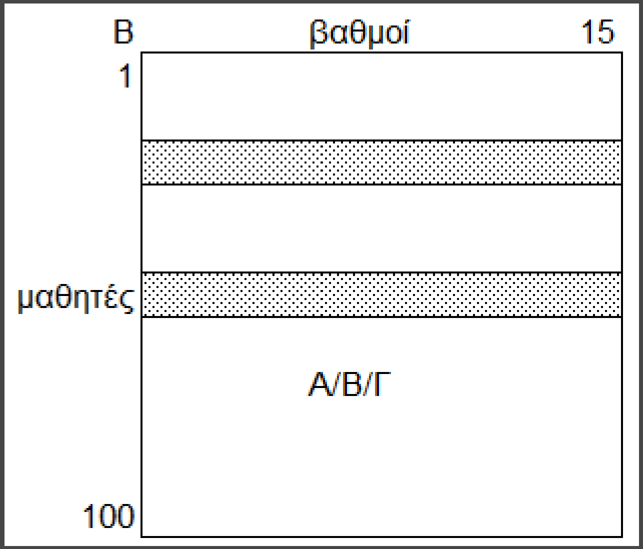



```
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν A[j-1, 1] > A[j, 1] τότε
      tmp <-- A[j, 1]
      A[j, 1] <-- A[j-1, 1]
      A[j-1, 1] <-- tmp
      tmp <-- A[j, 2]
      A[j, 2] <-- A[j-1, 2]
      A[j-1, 2] <-- tmp
    ΑλλιώςΑν A[j-1, 1] = A[j, 1]
ΚΑΙ A[j-1, 2] > A[j, 2] τότε
      tmp <-- A[j, 2]
      A[j, 2] <-- A[j-1, 2]
      A[j-1, 2] <-- tmp
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Γραμμές/Στήλες με ίδιες τιμές

Δίνεται ο B[100, 15] χαρακτήρων με τις επιδόσεις (Α/Β/Γ) 100 μαθητών σε 15 δοκιμασίες. Να ελεγχθεί και να εμφανίζεται κατάλληλο μήνυμα εάν υπάρχουν 2 μαθητές με τις ίδιες επιδόσεις και στις 15 δοκιμασίες καθώς και τους αριθμούς τους.

```
Π <-- 0
για i1 από 1 μέχρι 99
  για i2 από i1+1 μέχρι 100
    ΠΕ <-- 0
    για j από 1 μέχρι 15
      Αν B[i1, j] = B[i2, j] τότε
        ΠΕ <-- ΠΕ + 1
      ΤέλοςΑν
    ΤέλοςΕπανάληψης
    Αν ΠΕ=15 τότε
      Γράψε i1, i2
      Π <-- Π + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Αν Π <> 0 τότε
  Γράψε 'ναι'
Αλλιώς
  Γράψε 'όχι'
ΤέλοςΑν
```



Γραμμή/Στήλη με ίδιες τιμές

Δίνεται ο B[100, 15] χαρακτήρων με τις επιδόσεις (Α/Β/Γ) 100 μαθητών σε 15 δοκιμασίες. Να ελεγχθεί και να εμφανίζεται κατάλληλο μήνυμα εάν υπάρχει μαθητής με την ίδια επίδοση και στις 15 δοκιμασίες καθώς και τον αριθμό του.

```
Π <-- 0
για i από 1 μέχρι 100
  ΠΑ <-- 0 ΠΒ <-- 0 ΠΓ <-- 0
  για j από 1 μέχρι 15
    Αν B[i,j] = 'Α' τότε
      ΠΑ <-- ΠΑ + 1
    ΑλλιώςΑν B[i,j] = 'Β' τότε
      ΠΒ <-- ΠΒ + 1
    Αλλιώς
      ΠΓ <-- ΠΓ + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  Αν ΠΑ=15 Η ΠΒ=15 Η ΠΓ=15 τότε
    Γράψε i
    Π <-- Π + 1
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Αν Π <> 0 τότε
  Γράψε 'ναι'
Αλλιώς
  Γράψε 'όχι'
ΤέλοςΑν
```

Συμμετρία 2Δ πίνακα
Δίνεται ο A[80, 100]. Να ελεγχθεί αν είναι συμμετρικός αριστερά-δεξιά

1	6	11	16	16	11	6	1
2	7	12	17	17	12	7	2
3	8	13	18	18	13	8	3
4	9	14	19	19	14	9	4
5	10	15	20	20	15	10	5

```
συμ <-- Αληθής
για i από 1 μέχρι 80
  για j από 1 μέχρι 50
    Αν A[i, j] <> A[i, 101-j] τότε
      συμ <-- Ψευδής
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε συμ
```

A	1	2	3
1	1	2	3
2	4	5	6

Σαρώσεις 2-Δ πίνακα
Έστω ο παραπάνω πίνακας A[2, 3].
Συμπληρώστε τα κενά:

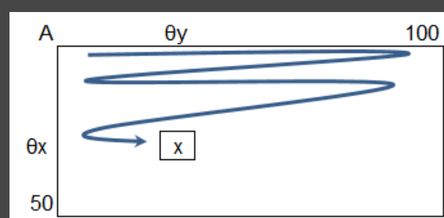
```
για ... από ... μέχρι ... μεβήμα ...
  για ... από ... μέχρι ... μεβήμα ...
    Γράψε A[i, j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

ώστε να εμφανισθούν
οι τιμές:
1) 1,2,3,4,5,6
2) 3,2,1,6,5,4
3) 4,5,6,1,2,3
4) 6,5,4,3,2,1
5) 1,4,2,5,3,6
6) 4,1,5,2,6,3
7) 3,6,2,5,1,4
8) 6,3,5,2,4,1

Απάντηση:
1) i, 1, 2, 1, j, 1, 3, 1
2) i, 1, 2, 1, j, 3, 1, -1
3) i, 2, 1, -1, j, 1, 3, 1
4) i, 2, 1, -1, j, 3, 1, -1
5) j, 1, 3, 1, i, 1, 2, 1
6) j, 1, 3, 1, i, 2, 1, -1
7) j, 3, 1, -1, i, 1, 2, 1
8) j, 3, 1, -1, i, 2, 1, -1

Σειριακή αναζήτηση – εντοπισμός της 1ης εμφάνισης σε 2Δ πίνακα *(σάρωση κατά γραμμές)*

```
Διάβασε x
βρ <-- Ψευδής
i <-- 1
Όσο i <= 50 ΚΑΙ βρ = Ψευδής επανάλαβε
  j <-- 1
  Όσο j <= 100 ΚΑΙ βρ = Ψευδής επανάλαβε
    Αν A[i,j] = x τότε
      βρ <-- Αληθής
      θx <-- i
      θy <-- j
    ΤέλοςΑν
    j <-- j + 1
  ΤέλοςΕπανάληψης
  i <-- i + 1
ΤέλοςΕπανάληψης
Αν βρ = Αληθής τότε
  Γράψε 'Βρέθηκε στη θέση:', θx, θy
Αλλιώς
  Γράψε 'Δεν βρέθηκε'
ΤέλοςΑν
```



"Τοπικό ελάχιστο"

Δίνεται ο ακέραιος A[50, 100]. Να μετρηθούν τα "τοπικά ελάχιστα" που υπάρχουν. Τα εσωτερικά δηλαδή στοιχεία που είναι μικρότερα και από τα 8 γειτονικά τους (οριζόντια/κάθετα/διαγώνια)

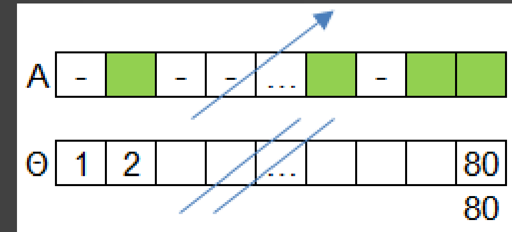
```
Π <-- 0
! σάρωση του A[50, 100] εντός της περιφέρειας
για i από 2 μέχρι 49
  για j από 2 μέχρι 99
    localmin <-- Αληθής
    ! σάρωση του 3x3 υπο-πίνακα με κέντρο το A[i, j]
    για x από i-1 μέχρι i+1
      για y από j-1 μέχρι j+1
        ! εξαίρεση του A[i, j]
        Αν (x<>i Η j<>y) ΚΑΙ A[x,y]<=A[i,j] τότε
          localmin <-- Ψευδής
        ΤέλοςΑν
      ΤέλοςΕπανάληψης
    ΤέλοςΕπανάληψης
    Αν localmin = Αληθής τότε
      Π <-- Π + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε Π
```

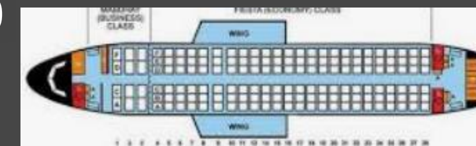
i-1,j-1	i-1,j	i-1,j+1
i,j-1	i,j	i,j+1
i+1,j-1	i+1,j	i+1,j+1

"1ο διαθέσιμο"

Αεροπορική πτήση διαθέτει 80 θέσεις επιβατών. Οι κρατήσεις των θέσεων γίνονται μέσω του πίνακα χαρακτήρων Α[80] με καταχώρηση της τιμής '-' για τις διαθέσιμες και του ονόματος του επιβάτη για τις κλεισμένες. Να γράψετε πρόγραμμα το οποίο αρχικοποιεί τον Α[80] με '-'. Μέσω επαναληπτικού μενού εκτελεί τα παρακάτω:

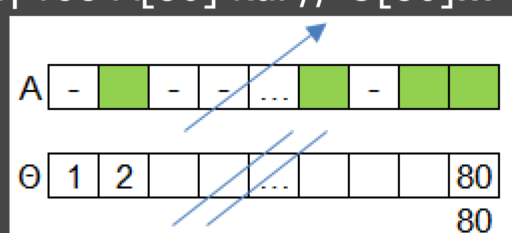
1. Αγορά: βρίσκει την 1η διαθέσιμη θέση και διαβάζει σε αυτήν το όνομα του κατόχου. Αν δεν υπάρχει διαθέσιμη θέση, εμφανίζει κατάλληλο μήνυμα.
2. Αντικατάσταση: αν υπάρχει κλεισμένη θέση διαβάζει με έλεγχο εγκυρότητας έναν αριθμό θέσης (1-80) που είναι και αριθμός κλεισμένης θέσης και διαβάζει σε αυτήν το όνομα του νέου κατόχου. Αν δεν υπάρχει κλεισμένη θέση, εμφανίζει κατάλληλο μήνυμα.
3. Ακύρωση: αν υπάρχει κλεισμένη θέση διαβάζει με έλεγχο εγκυρότητας έναν αριθμό θέσης (1-80) που είναι και αριθμός κλεισμένης θέσης και εκχωρεί σε αυτήν την τιμή '-'. Αν δεν υπάρχει κλεισμένη θέση, εμφανίζει κατάλληλο μήνυμα.
4. Τερματισμός: εμφανίζει αλφαβητικά τους επιβάτες της πτήσης μαζί με τον αριθμό της θέσης τους καθώς και το πλήθος τους.





```
για i από 1 μέχρι 80
  Α[i] <-- '-'
  Θ[i] <-- i
ΤέλοςΕπανάληψης
Π <-- 0 ! πλήθος επιβατών
ΑρχήΕπανάληψης
  Γράψε '1. Αγορά'
  Γράψε '2. Αντικατάσταση'
  Γράψε '3. Ακύρωση'
  Γράψε '4. Τερματισμός'
  Διάβασε επ
  Αν επ = 1 τότε ! Αγορά
    Αν Π < 80 τότε
      i <-- 1
      Όσο Α[i] <> '-' επανάλαβε
        i <-- i + 1
      ΤέλοςΕπανάληψης
      Διάβασε Α[i]
      Π <-- Π + 1
    Αλλιώς
      Γράψε 'Καμία διαθέσιμη'
    ΤέλοςΑν
```

```
 ΑλλιώςΑν επ = 2 τότε ! Αντικατάσταση
   Αν Π > 0 τότε
     ΑρχήΕπανάληψης
       Διάβασε x
     ΜέχριςΌτου x>=1 ΚΑΙ x<=80 ΚΑΙ A[x]<>'-'
     Διάβασε A[x]
   Αλλιώς
     Γράψε 'Καμία κλεισμένη'
   ΤέλοςΑν
 ΑλλιώςΑν επ = 3 τότε ! Ακύρωση
   Αν Π > 0 τότε
     ΑρχήΕπανάληψης
       Διάβασε x
     ΜέχριςΌτου x>=1 ΚΑΙ x<=80 ΚΑΙ A[x]<>'-'
     A[x] ← '-'
     Π <-- Π - 1
   Αλλιώς
     Γράψε 'Καμία κλεισμένη'
   ΤέλοςΑν
 ΤέλοςΑν
ΜέχριςΌτου επ = 4
! αύξουσα ταξινόμηση του A[80] και // Θ[80]...
για i από 1 μέχρι 80
  Αν A[i] <> '-' τότε
    Γράψε A[i], Θ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
Γράψε 'Σύνολο: ', Π
```



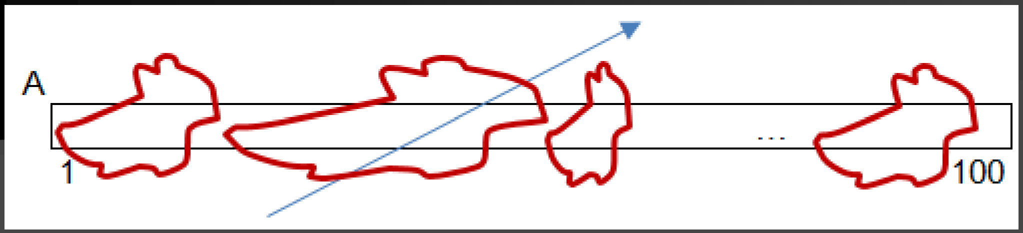
A	2	1	4	5	3
B	1	2	5	3	4

Συμπληρώστε το κενό ώστε αν ο πίνακας A έχει τις παραπάνω τιμές, στον B να καταχωρηθούν οι αντίστοιχες:

```
για i από 1 μέχρι 5
  B[i] ← A[.................]
ΤέλοςΕπανάληψης
```

"Συχνότερες" τιμές συνόλου γνωστού πλήθους

Να γραφεί πρόγραμμα το οποίο διαβάζει τον ακέραιο A[100] και εμφανίζει τις "συχνότερες" τιμές του.



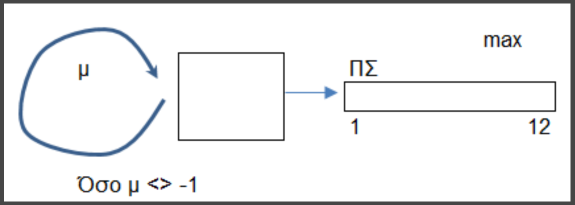

```
! εισαγωγή πίνακα
για i από 1 μέχρι 100
  Διάβασε A[i]
ΤέλοςΕπανάληψης
! ταξινόμηση πίνακα
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν A[j-i]>A[j] τότε
      tmp <-- A[j-1]
      A[j-1] <-- A[j]
      A[j] <-- tmp
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

```
! υπολογισμός μέγιστου σερί
seri <-- 1
maxseri <-- 1
για i από 2 μέχρι 100
  Αν A[i] = A[i-1] τότε
    seri <-- seri + 1
  Αλλιώς
    seri <-- 1
  ΤέλοςΑν
  Αν seri > maxseri τότε
    maxseri <-- seri
  ΤέλοςΑν
ΤέλοςΕπανάληψης
! αναζήτηση μέγιστου σερί
seri <-- 1
για i από 2 μέχρι 100
  Αν A[i] = A[i-1] τότε
    seri <-- seri + 1
  Αλλιώς
    seri <-- 1
  ΤέλοςΑν
  Αν seri = maxseri τότε
    Γράψε A[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

"Συχνότερες" τιμές συνόλου αγνώστου πλήθους

Να γραφεί πρόγραμμα το οποίο διαβάζει επαναληπτικά ακέραιες τιμές στο διάστημα 1-12 για μία δημοσκόπηση "ποιος είναι ο αγαπημένος σας μήνας του έτους" μέχρι να δοθεί ως μήνας η τιμή -1. Να εμφανίζει τους πιο αγαπημένους μήνες.

```
για i από 1 μέχρι 12
  ΠΣ[i] <-- 0
ΤέλοςΕπανάληψης
Διάβασε μ
Όσο μ <> -1 επανάλαβε
  ΠΣ[μ] <-- ΠΣ[μ] + 1
  Διάβασε μ
ΤέλοςΕπανάληψης
max <-- ΠΣ[1]
για i από 2 μέχρι 12
  Αν ΠΣ[i] > max τότε
    max <-- ΠΣ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 12
  Αν ΠΣ[i] = max τότε
    Γράψε i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```



"Φεύγει ο μικρότερος"

Σε έναν διαγωνισμό συμμετέχουν 100 υποψήφιοι. Σε κάθε γύρο λαμβάνει ο καθένας μία βαθμολογία και αποχωρεί αυτός με τη μικρότερη (έστω ένας κάθε φορά). Νικητής είναι αυτός που θα παραμείνει τελευταίος στον διαγωνισμό. Να γραφεί πρόγραμμα που α) διαβάζει τα ονόματα των 100 υποψηφίων β) για κάθε γύρο διαβάζει τους βαθμούς (0-100) όσων έχουν απομείνει και αποκλείει τον χειρότερο γ) εμφανίζει τον νικητή.

```
για i από 1 μέχρι 100
  Διάβασε ΟΝ[i]
ΤέλοςΕπανάληψης
για i από 1 μέχρι 99
  min <-- 101
  για j από 1 μέχρι 100
    Αν ΟΝ[j] <> '-' τότε
      Διάβασε β
      Αν β < min τότε
        min <-- β
        θmin <-- j
      ΤέλοςΑν
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  ΟΝ[θmin] <-- '-'
ΤέλοςΕπανάληψης
```

```
για i από 1 μέχρι 100
  Αν ΟΝ[i] <> '-' τότε
    Γράψε ΟΝ[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Μέγιστο σερί κατά γραμμές 2Δ πίνακα

Να γραφεί πρόγραμμα το οποίο διαβάζει α) στον B[50, 30] τις εταιρείες αυτοκινήτων που βραβεύτηκαν τα τελευταία 50 χρόνια σε 30 κατηγορίες αξιολόγησης (ασφάλεια, οικονομία κλπ.) β) εμφανίζει τον μεγαλύτερο αριθμό βραβείων που κέρδισε εταιρεία σε μία χρονιά γ) εμφανίζει τις εταιρείες που κέρδισαν τον μεγαλύτερο αριθμό βραβείων σε μία χρονιά.

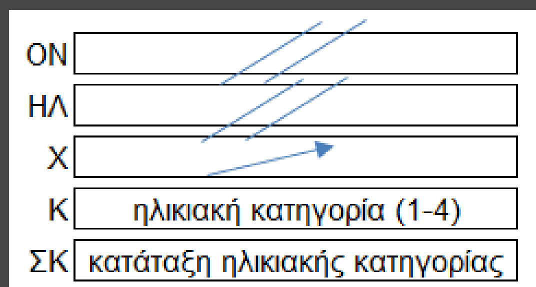

```
! εισαγωγή
για i από 1 μέχρι 50
  για j από 1 μέχρι 30
    Διάβασε B[i,j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

```
! ταξινόμηση των γραμμών
για i από 1 μέχρι 50
  για κ από 2 μέχρι 30
    για λ από 30 μέχρι κ μεβήμα -1
      Αν B[i, λ-1] > B[i, λ] τότε
        tmp <-- B[i, λ-1]
        B[i, λ-1] <-- B[i, λ]
        B[i, λ] <-- tmp
      ΤέλοςΑν
    ΤέλοςΕπανάληψης
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

```
! υπολογισμός του μέγιστου σερί
maxseri <-- 0
για i από 1 μέχρι 50
  σερί <-- 1
  για j από 2 μέχρι 30
    Αν B[i, j] = B[i, j-1] τότε
      σερί <-- σερί + 1
    Αλλιώς
      σερί <-- 1
    ΤέλοςΑν
    Αν σερί > maxseri τότε
      maxseri <-- σερί
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
Γράψε maxseri
! αναζήτηση του μέγιστου σερί
για i από 1 μέχρι 50
  σερί <-- 1
  για j από 2 μέχρι 30
    Αν B[i, j] = B[i, j-1] τότε
      σερί <-- σερί + 1
    Αλλιώς
      σερί <-- 1
    ΤέλοςΑν
    Αν σερί = maxseri τότε
      Γράψε B[i, j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Σειρά κατάταξης

Σε έναν αγώνα δρόμου συμμετείχαν 3000 δρομείς. Με βάση την ηλικία τους χωρίζονται σε 4 κατηγορίες: κάτω των 20, των 40, των 60 και οι μεγαλύτεροι. Να γραφεί πρόγραμμα το οποίο α) διαβάζει σε πίνακες τα ονόματα, τις ηλικίες και τους χρόνους τερματισμού (sec) των δρομέων β) υπολογίζει για τον καθένα τη σειρά τερματισμού στη γενική κατάταξη και στην ηλικιακή κατηγορία που ανήκει γ) εμφανίζει τα παραπάνω με σειρά γενικής κατάταξης.



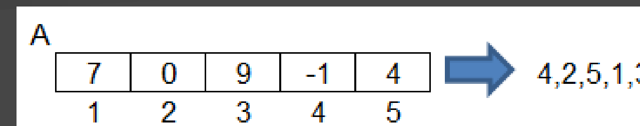

```
για i από 1 μέχρι 3000
  Διάβασε ΟΝ[i], ΗΛ[i], Χ[i]
ΤέλοςΕπανάληψης
! αύξουσα ταξινόμηση του
! Χ[3000] και // ΟΝ,ΗΛ...
! ηλικιακές κατηγορίες
για i από 1 μέχρι 3000
  Αν ΗΛ[i] < 20 τότε
    Κ[i] <-- 1
  ΑλλιώςΑν ΗΛ[i] < 40 τότε
    Κ[i] <-- 2
  ΑλλιώςΑν ΗΛ[i] < 60 τότε
    Κ[i] <-- 3
  Αλλιώς
    Κ[i] <-- 4
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

```
! σειρές ανά κατηγορία
για i από 1 μέχρι 3000
  ΣΚ[i] <-- 1
  για j από 1 μέχρι i-1
    Αν Χ[j]<Χ[i] ΚΑΙ Κ[i]=Κ[j] τότε
      ΣΚ[i] <-- ΣΚ[i] + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
! αποτελέσματα
για i από 1 μέχρι 3000
  Γράψε i, ΟΝ[i], Χ[i], ΣΚ[i], Κ[i]
ΤέλοςΕπανάληψης
```

Ταξινόμηση δεικτών

Να γραφεί πρόγραμμα που διαβάζει ακέραιο πίνακα Α[100] και εμφανίζει τους δείκτες των στοιχείων του έτσι ώστε οι τιμές τους να είναι ταξινομημένες κατά αύξουσα σειρά.

π.χ.




```
για i από 1 μέχρι 100
  Διάβασε Α[i]
  Δ[i] <-- i
ΤέλοςΕπανάληψης
για i από 2 μέχρι 100
  για j από 100 μέχρι i μεβήμα -1
    Αν Α[j-i]>A[j] τότε
      tmp <-- A[j-1]
      A[j-1] <-- A[j]
      A[j] <-- tmp
      tmp2 <-- Δ[j-1]
      Δ[j-1] <-- Δ[j]
      Δ[j] <-- tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 100
  Γράψε Δ[i]
ΤέλοςΕπανάληψης
```

Διοργάνωση συλλόγων

Με μονούς αγώνες

Σε μία ποδοσφαιρική διοργάνωση συμμετέχουν 10 ομάδες. Κάθε ομάδα παίζει ένα αγώνα με καθεμιά από τις υπόλοιπες ομάδες. Πρόγραμμα που διαβάζει α) σε πίνακα O[10] τα ονόματα των ομάδων β) για κάθε αγώνα το σκορ του. Να ενημερώνει τον βαθμολογικό πίνακα B[10]

```
για i από 1 μέχρι 10
  Διάβασε O[i]
  B[i] <-- 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 9
  για j από i+1 μέχρι 10
    Γράψε O[i], O[j]
    Διάβασε γ1, γ2
    Αν γ1 > γ2 τότε
      B[i] <-- B[i] + 3
    ΑλλιώςΑν γ2 > γ1 τότε
      B[j] <-- B[j] + 3
    Αλλιώς
      B[i] <-- B[i] + 1
      B[j] <-- B[j] + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

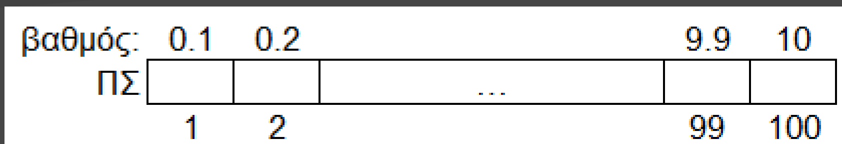
Με διπλούς αγώνες

Σε μία ποδοσφαιρική διοργάνωση συμμετέχουν 10 ομάδες. Κάθε ομάδα παίζει δύο αγώνες με καθεμιά από τις υπόλοιπες ομάδες. Πρόγραμμα που διαβάζει α) σε πίνακα O[10] τα ονόματα των ομάδων β) για κάθε αγώνα το σκορ του. Να ενημερώνει τον βαθμολογικό πίνακα B[10]

```
για i από 1 μέχρι 10
  Διάβασε O[i]
  B[i] <-- 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 10
  για j από 1 μέχρι 10
    Αν i<>j τότε
      Γράψε O[i], O[j]
      Διάβασε γ1, γ2
      Αν γ1 > γ2 τότε
        B[i] <-- B[i] + 3
      ΑλλιώςΑν γ2 > γ1 τότε
        B[j] <-- B[j] + 3
      Αλλιώς
        B[i] <-- B[i] + 1
        B[j] <-- B[j] + 1
      ΤέλοςΑν
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
```

Πίνακας συχνοτήτων πραγματικών τιμών

Πρόγραμμα που διαβάζει τους βαθμούς 300 φοιτητών σε ένα μάθημα με έλεγχο ώστε να ανήκουν στο [0.1, 10] και με ακρίβεια ενός το πολύ δεκαδικού ψηφίου. Να εμφανίζει την % συχνότητα εμφάνισης του κάθε βαθμού: 0.1, 0.2, ..., 9.9, 10




```
για i από 1 μέχρι 100
  ΠΣ[i] <-- 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 300
  ΑρχήΕπανάληψης
    Διάβασε β
    δμβ <-- β - Α_Μ(β)
  ΜέχριςΌτου β >= 0.1 ΚΑΙ β <= 10 ΚΑΙ
δμβ*10 = Α_Μ(δμβ*10)
  x <-- Α_Μ(β*10)
  ΠΣ[x] <-- ΠΣ[x] + 1
ΤέλοςΕπανάληψης
για i από 1 μέχρι 100
  β <-- i/10
  Γράψε 'Βαθμός ', β, ' % συχνότητα ',
ΠΣ[i]/300*100
ΤέλοςΕπανάληψης
```

Σειριακή αναζήτηση Ν-στής εμφάνισης

Πρόγραμμα που διαβάζει μία ακέραια τιμή x και την αναζητάει στον ακέραιο Α[100] μέχρι να τη βρει 5 φορές. Εμφανίζει τη θέση της 5ης εμφάνισης του x ή κατάλληλο μήνυμα αν δεν βρεθεί 5 φορές

```
Διάβασε x
πβρ <-- 0
i <-- 1
Όσο i <= 100 ΚΑΙ πβρ < 5 επανάλαβε
  Αν (x = A[i]) τότε
    πβρ <-- πβρ + 1
    θέση <-- i
  ΤέλοςΑν
  i <-- i + 1
ΤέλοςΕπανάληψης
Αν πβρ = 5 τότε
  Γράψε 'Βρέθηκε για 5η φορά στο κελί: ', θέση
Αλλιώς
  Γράψε 'Δεν βρέθηκε 5 φορές'
ΤέλοςΑν
```

Συμπληρώστε το κενό ώστε σε κάθε περίπτωση να καταχωρούνται σε πραγματικό πίνακα Α[5,5] ή σε λογικό πίνακα Λ[5,5] (Α:Αληθής, Ψ:Ψευδής), οι παρακάτω τιμές:

για i από 1 μέχρι 5
  για j από 1 μέχρι 5
    Α[i,j] ή Λ[i, j] ← ......
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης

Α 17

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

Α 1

2	3	4	5	6
3	4	5	6	7
4	5	6	7	8
5	6	7	8	9
6	7	8	9	10

Α 2

0	-1	-2	-3	-4
1	0	-1	-2	-3
2	1	0	-1	-2
3	2	1	0	-1
4	3	2	1	0

Α 3

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Α 4

1	0.5	0.33	0.25	0.2
2	1	0.67	0.5	0.4
3	1.5	1	0.75	0.6
4	2	1.33	1	0.8
5	2.5	1.67	1.25	1

Α 5

1	1	1	1	1
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

Α 6

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

Α 7

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

Α 8

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

Α 9

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Α 10

25	24	23	22	21
20	19	18	17	16
15	14	13	12	11
10	9	8	7	6
5	4	3	2	1

Α 11

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

Α 12

25	20	15	10	5
24	19	14	9	4
23	18	13	8	3
22	17	12	7	2
21	16	11	6	1

Α 13

1.1	1.2	1.3	1.4	1.5
2.1	2.2	2.3	2.4	2.5
3.1	3.2	3.3	3.4	3.5
4.1	4.2	4.3	4.4	4.5
5.1	5.2	5.3	5.4	5.5

Α 14

1	1.5	2	2.5	3
1.5	2	2.5	3	3.5
2	2.5	3	3.5	4
2.5	3	3.5	4	4.5
3	3.5	4	4.5	5

Λ 15

Ψ	Α	Α	Α	Α
Ψ	Ψ	Α	Α	Α
Ψ	Ψ	Ψ	Α	Α
Ψ	Ψ	Ψ	Ψ	Α
Ψ	Ψ	Ψ	Ψ	Ψ

Λ 16

Α	Ψ	Ψ	Ψ	Ψ
Ψ	Α	Ψ	Ψ	Ψ
Ψ	Ψ	Α	Ψ	Ψ
Ψ	Ψ	Ψ	Α	Ψ
Ψ	Ψ	Ψ	Ψ	Α

Απαντήσεις	
1	i+j
2	i-j
3	i*j
4	i/j
5	i^j
6	i div j
7	i mod j
8	i*10+j
9	(i-1)*5+j
10	26-((i-1)*5+j)
11	(j-1)*5+i
12	26-((j-1)*5+i)
13	i+j/10
14	(i+j)/2
15	i < j
16	i = j
17	(i+j) mod 2

Τηλεπαιχνίδι

Το "Ποιος Θέλει να Γίνει Εκατομμυριούχος" είναι τηλεπαιχνίδι γνώσεων. Ο παίκτης πρέπει να απαντήσει σωστά σε 15 ερωτήσεις για να κερδίσει 100.000€. Για κάθε ερώτηση, υπάρχουν τέσσερις πιθανές απαντήσεις, με μία μόνο σωστή. Κάθε ερώτηση έχει τη δική της χρηματική αξία που φαίνεται στον πίνακα. Τα παρακάτω ποσά δεν είναι αθροιστικά. Π.χ. αν ο παίκτης απαντήσει σωστά στην δεύτερη ερώτηση, έχει κερδίσει συνολικά 200€ και όχι 100€+200€. Το παιχνίδι για τον κάθε παίκτη μπορεί τελειώσει με 3 πιθανούς τρόπους:

α) Εάν δώσει μία λανθασμένη απάντηση οπότε είτε φεύγει με άδεια χέρια (αν έχει χάσει σε μία από τις πρώτες 5 ερωτήσεις) είτε κερδίζει το μικρό ή το μεγάλο μαξιλαράκι (αν έχει καταφέρει προηγουμένως να απαντήσει σωστά στην 5η ή 10η ερώτηση αντίστοιχα)

β) Εάν αποφασίσει να σταματήσει οπότε κερδίζει το ποσό της τελευταίας ερώτησης που απάντησε σωστά

γ) Εάν καταφέρει να απαντήσει σωστά σε όλες τις ερωτήσεις οπότε κερδίζει το μέγιστο χρηματικό έπαθλο

Να γραφεί πρόγραμμα το οποίο: α) καταχωρεί στον πίνακα ΑΞΙΑ[15] (ακέραιος) τα ποσά των ερωτήσεων β) διαβάζει στους παρακάτω πίνακες i. ΕΡ[15] (χαρακτήρων) τις 15 ερωτήσεις ii. ΑΠ[15, 4] (χαρακτήρων) τις τέσσερις πιθανές απαντήσεις κάθε ερώτησης iii. Σ[15] (ακέραιος) τις σωστές απαντήσεις (1-4) με έλεγχο εγκυρότητας γ) επαναληπτικά: i. εμφανίζει την κάθε ερώτηση, το ποσό της και τις τέσσερις πιθανές απαντήσεις ii. διαβάζει την επιλογή του παίκτη (1-4) αν απάντησε ή την τιμή 0 αν αποφασίσει να σταματήσει δ) τελικά εμφανίζει ένα από τα παρακάτω μηνύματα: i. "Next time" αν δεν κέρδισε τίποτα ii. "Κέρδισες Χ€!" αν κέρδισε κάποιο ποσό πριν το μέγιστο iii. "Τα πήρες όλα!!!" αν κέρδισε το μέγιστο ποσό.

Ερώτηση	15	100,000 €	
Ερώτηση	14	80,000 €	
Ερώτηση	13	60,000 €	
Ερώτηση	12	40,000 €	
Ερώτηση	11	20,000 €	
Ερώτηση	10	5,000 €	(μαξιλάρι)
Ερώτηση	9	4,000 €	
Ερώτηση	8	3,000 €	
Ερώτηση	7	2,000 €	
Ερώτηση	6	1,000 €	
Ερώτηση	5	500 €	(μαξιλάρι)
Ερώτηση	4	400 €	
Ερώτηση	3	300 €	
Ερώτηση	2	200 €	
Ερώτηση	1	100 €	

Τηλεπαιχνίδι

```
για i από 1 μέχρι 5 ! α)
  ΑΞΙΑ[i] <-- 100*i
  ΑΞΙΑ[i+5] <-- 1000*i
  ΑΞΙΑ[i+10] <-- 20000*i
ΤέλοςΕπανάληψης
για i από 1 μέχρι 15 ! β)
  Διάβασε ΕΡ[i]
  για j από 1 μέχρι 4
    Διάβασε ΑΠ[i, j]
  ΤέλοςΕπανάληψης
  ΑρχήΕπανάληψης
    Διάβασε Σ[i]
  ΜέχριςΌτου Σ[i] >= 1 ΚΑΙ Σ[i] <= 4
ΤέλοςΕπανάληψης
στοπ <-- Ψευδής ! γ)
σταμάτησε <-- Ψευδής
π <-- 1
ΑρχήΕπανάληψης
  Γράψε ΕΡ[π], ΑΞΙΑ[π] ! γ.i
  για j από 1 μέχρι 4
    Γράψε ΑΠ[π, j]
  ΤέλοςΕπανάληψης
  Γράψε 'Δώσε 1-4 για απάντηση ή 0 για διακοπή'
  ΑρχήΕπανάληψης ! γ.ii
    Διάβασε επ
  ΜέχριςΌτου επ >= 0 ΚΑΙ επ <= 4
Αν επ = 0 τότε
    σταμάτησε <-- Αληθής
    στοπ <-- Αληθής
  ΑλλιώςΑν επ <> Σ[π] τότε
    στοπ <-- Αληθής
  Αλλιώς
    π <-- π + 1
    Αν π = 16 τότε
      στοπ <-- Αληθής
    ΤέλοςΑν
  ΤέλοςΑν
ΜέχριςΌτου στοπ = Αληθής
Αν π = 16 τότε  ! δ)
  Γράψε "Τα πήρες όλα!!!"
ΑλλιώςΑν σταμάτησε = Αληθής τότε
  Αν π > 1 τότε
    Γράψε 'Κέρδισες ', ΑΞΙΑ[π-1], '€!'
  Αλλιώς
    Γράψε 'Next time'
  ΤέλοςΑν
Αλλιώς ! έχασε στην π ερώτηση
  Αν π > 10 τότε
    Γράψε 'Κέρδισες ', ΑΞΙΑ[10], '€!'
  ΑλλιώςΑν π > 5 τότε
    Γράψε 'Κέρδισες ', ΑΞΙΑ[5], '€!'
  Αλλιώς
    Γράψε 'Next time'
  ΤέλοςΑν
ΤέλοςΑν
```

	ΑΞΙΑ	ΕΡ	Σ	ΑΠ 1	2	3	4
1							
15							

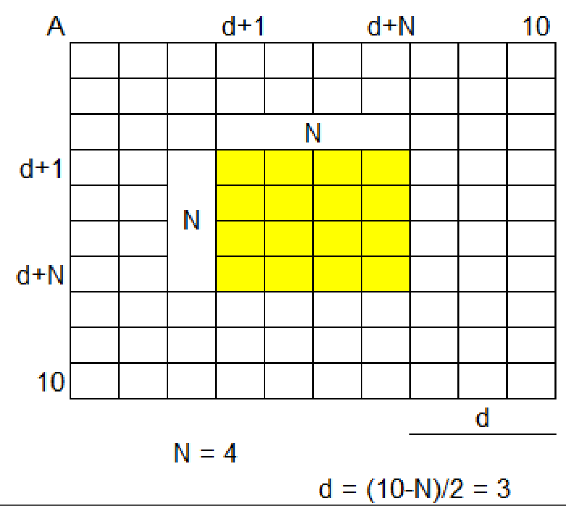
Ερώτηση	15	100,000 €	
Ερώτηση	14	80,000 €	
Ερώτηση	13	60,000 €	
Ερώτηση	12	40,000 €	
Ερώτηση	11	20,000 €	
Ερώτηση	10	5,000 €	(μαξιλάρι)
Ερώτηση	9	4,000 €	
Ερώτηση	8	3,000 €	
Ερώτηση	7	2,000 €	
Ερώτηση	6	1,000 €	
Ερώτηση	5	500 €	(μαξιλάρι)
Ερώτηση	4	400 €	
Ερώτηση	3	300 €	
Ερώτηση	2	200 €	
Ερώτηση	1	100 €	

Σειριακή αναζήτηση 2 τιμών σε A[100]

```
Διάβασε x1, x2
βρ1 <-- Ψευδής
βρ2 <-- Ψευδής
i <-- 1
Όσο i <= 100 ΚΑΙ (βρ1 = Ψευδής Η βρ2 = Ψευδής)
επανάλαβε
  Αν x1 = A[i] τότε
    βρ1 <-- Αληθής
    θέση1 <-- i
  ΤέλοςΑν
  Αν x2 = A[i] τότε
    βρ2 <-- Αληθής
    θέση2 <-- i
  ΤέλοςΑν
  i <-- i + 1
ΤέλοςΕπανάληψης
Αν βρ1 = Αληθής τότε
  Γράψε 'Βρέθηκε 1η τιμή στο κελί: ', θέση1
Αλλιώς
  Γράψε 'Δεν βρέθηκε η 1η τιμή’
ΤέλοςΑν
Αν βρ2 = Αληθής τότε
  Γράψε 'Βρέθηκε 2η τιμή στο κελί: ', θέση2
Αλλιώς
  Γράψε 'Δεν βρέθηκε η 2η τιμή'
ΤέλοςΑν
```

Κεντραρισμένος υποπίνακας

Πρόγραμμα που διαβάζει ακέραιο A[100, 100] και έναν θετικό ακέραιο Ν με εγκυρότητα ώστε να είναι άρτιος και το πολύ 100. Να εμφανίζει το άθροισμα του NxN υποπίνακα του A που είναι κεντραρισμένος στον A.




```
για i από 1 μέχρι 100
  για j από 1 μέχρι 100
    Διάβασε A[i,j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
ΑρχήΕπανάληψης
  Διάβασε N
ΜέχριςΌτου N > 0 ΚΑΙ N <=
100 ΚΑΙ N mod 2 = 0
d <-- (100-N)/2
S <-- 0
για i από d+1 μέχρι d+N
  για j από d+1 μέχρι d+N
    S <-- S + A[i,j]
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
ΑρχήΕπανάληψης
Γράψε S
```

Παιχνίδι Ν παικτών

Πρόγραμμα που διαβάζει τα ονόματα 30 παικτών ενός παιχνιδιού ρίψης ενός ζαριού. Σε κάθε γύρο του παιχνιδιού οι παίκτες ρίχνουν το ζάρι και κερδίζουν έναν πόντο όσοι έφεραν τη μεγαλύτερη τιμή. Για κάθε γύρο του παιχνιδιού να διαβάζει τις τιμές του ζαριού για τους 30 παίκτες. Να εμφανίζει την τελική κατάταξη **στους 20 γύρους**.

```
για i από 1 μέχρι 30 ! α)
  Διάβασε Ο[i]
  Π[i] <-- 0
ΤέλοςΕπανάληψης
για i από 1 μέχρι 20 ! για κάθε γύρο
  για j από 1 μέχρι 30 ! για κάθε παίκτη
    ΑρχήΕπανάληψης
      Διάβασε Z[j]
    ΜέχριςΌτου Z[j] >= 1 ΚΑΙ Z[j] <= 6
  ΤέλοςΕπανάληψης
  max <-- Z[1]
  για j από 2 μέχρι 30
    Αν Z[j] > max τότε
      max <-- Z[j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  για j από 1 μέχρι 30
    Αν Z[j] = max τότε
      Π[j] <-- Π[j] + 1
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 2 μέχρι 30
  για j από 30 μέχρι i μεβήμα -1
    Αν Π[j-1] < Π[j] τότε
      tmp1 <-- Π[j-1]
      Π[j-1] <-- Π[j]
      Π[j] <-- tmp1
      tmp2 <-- Ο[j-1]
      Ο[j-1] <-- Ο[j]
      Ο[j] <-- tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 30
  Γράψε Ο[i], Π[i]
ΤέλοςΕπανάληψης
```

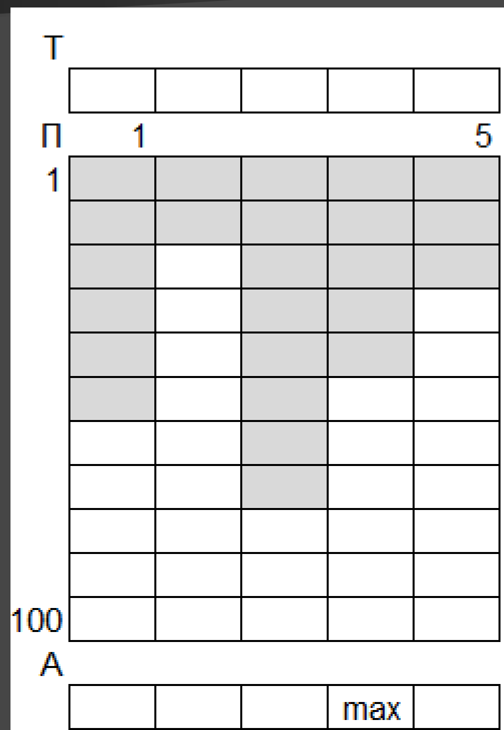
Παιχνίδι Ν παικτών

Πρόγραμμα που διαβάζει τα ονόματα 30 παικτών ενός παιχνιδιού ρίψης ενός ζαριού. Σε κάθε γύρο του παιχνιδιού οι παίκτες ρίχνουν το ζάρι και κερδίζουν έναν πόντο όσοι έφεραν τη μεγαλύτερη τιμή. Για κάθε γύρο του παιχνιδιού να διαβάζει τις τιμές του ζαριού για τους 30 παίκτες. Να εμφανίζει την τελική κατάταξη **στις 20 νίκες.**

```
για i από 1 μέχρι 30
  Διάβασε Ο[i]
  Π[i] <-- 0
ΤέλοςΕπανάληψης
end <-- Ψευδής
ΑρχήΕπανάληψης
  για j από 1 μέχρι 30 ! για κάθε παίκτη
    ΑρχήΕπανάληψης
      Διάβασε Ζ[j]
    ΜέχριςΌτου Ζ[j] >= 1 ΚΑΙ Ζ[j] <= 6
  ΤέλοςΕπανάληψης
  max <-- Z[1]
  για j από 2 μέχρι 30
    Αν Ζ[j] > max τότε
      max <-- Z[j]
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
  για j από 1 μέχρι 30
    Αν Ζ[j] = max τότε
      Π[j] <-- Π[j] + 1
      Αν Π[j] = 20 τότε
        end <-- Αληθής
      ΤέλοςΑν
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΜέχριςΌτου end = Αληθής
για i από 2 μέχρι 30
  για j από 30 μέχρι i μεβήμα -1
    Αν Π[j-1] < Π[j] τότε
      tmp1 <-- Π[j-1]
      Π[j-1] <-- Π[j]
      Π[j] <-- tmp1
      tmp2 <-- Ο[j-1]
      Ο[j-1] <-- Ο[j]
      Ο[j] <-- tmp2
    ΤέλοςΑν
  ΤέλοςΕπανάληψης
ΤέλοςΕπανάληψης
για i από 1 μέχρι 30
  Γράψε Ο[i], Π[i]
ΤέλοςΕπανάληψης
```

Στοίβες σε 2Δ πίνακα

Πρόγραμμα που διαχειρίζεται 5 ακέραιες στοίβες χωρητικότητας 100 σε 2Δ πίνακα Π[100, 5]. Οι δείκτες κορυφή βρίσκονται στον Τ[5]. Αρχικοποιεί τις στοίβες ως άδειες. Με κατάλληλο επαναληπτικό μενού εκτελεί τα παρακάτω: 1. Ώθηση: ωθεί μία εισαγόμενη τιμή στην πρώτη στοίβα με τα λιγότερα στοιχεία. 2. Απώθηση: διαβάζει με έλεγχο εγκυρότητας τον αριθμό μιας στοίβας (1-5) και απωθεί από αυτήν και εμφανίζει το κορυφαίο στοιχείο της. 3. Έξοδος: τερματίζει το πρόγραμμα εμφανίζοντας τη στοίβα όπου έγιναν οι περισσότερες απωθήσεις.




```
για i από 1 μέχρι 5
  Τ[i] <-- 0
  Α[i] <-- 0
ΤέλοςΕπανάληψης
ΑρχήΕπανάληψης
  Γράψε '1. Ώθηση'
  Γράψε '2. Απώθηση'
  Γράψε '3. Έξοδος'
  Διάβασε επ
  Αν επ = 1 τότε
    min <-- Τ[1]
    θmin <-- 1
    για i από 2 μέχρι 5
      Αν Τ[i] < min τότε
        min <-- Τ[i]
        θmin <-- i
      ΤέλοςΑν
    ΤέλοςΕπανάληψης
    Αν Τ[θmin] = 100 τότε
      Γράψε 'Υπερχείλιση'
    Αλλιώς
      Τ[θmin] <-- Τ[θmin] + 1
      Διάβασε Π[Τ[θmin], θmin]
    ΤέλοςΑν
  ΑλλιώςΑν επ = 2 τότε
    ΑρχήΕπανάληψης
      Διάβασε x
    ΜέχριςΌτου x >= 1 ΚΑΙ x <= 5
    Αν Τ[x] = 0 τότε
      Γράψε 'Υποχείλιση'
    Αλλιώς
      Γράψε Π[Τ[x], x]
      Τ[x] <-- Τ[x] - 1
      Α[x] <-- Α[x] + 1
    ΤέλοςΑν
  ΤέλοςΑν
ΜέχριςΌτου επ = 3
max <-- Α[1]
για i από 2 μέχρι 5
  Αν Α[i] > max τότε
    max <-- Α[i]
  ΤέλοςΑν
ΤέλοςΕπανάληψης
για i από 1 μέχρι 5
  Αν Α[i] = max τότε
    Γράψε i
  ΤέλοςΑν
ΤέλοςΕπανάληψης
```

Ουρές σε 2Δ πίνακα

Πρόγραμμα που διαχειρίζεται 5 ακέραιες ουρές χωρητικότητας 100 σε 2Δ πίνακα Q[5, 100]. Οι δείκτες εμπρός και πίσω βρίσκονται στους F[5] και R[5] αντίστοιχα. Αρχικοποιεί τις ουρές ως άδειες. Με κατάλληλο επαναληπτικό μενού εκτελεί τα παρακάτω: 1. Εισαγωγή: εισάγει μία εισαγόμενη τιμή στην πρώτη ουρά με τα λιγότερα στοιχεία. 2. Εξαγωγή: διαβάζει με έλεγχο εγκυρότητας τον αριθμό μιας ουράς (1-5) και εξάγει από αυτήν και εμφανίζει το μπροστινό στοιχείο της. 3. Έξοδος: τερματίζει το πρόγραμμα εμφανίζοντας την ουρά όπου έγιναν οι περισσότερες εξαγωγές.

```
για i από 1 μέχρι 5
  F[i] <-- 0   R[i] <-- 0  A[i] <-- 0
ΤέλοςΕπανάληψης
ΑρχήΕπανάληψης
  Γράψε '1. Εισαγωγή'
  Γράψε '2. Εξαγωγή'
  Γράψε '3. Έξοδος'
  Διάβασε επ
  Αν επ = 1 τότε
    Αν F[1] = 0 τότε
      min <-- 0
      θmin <-- 1
    Αλλιώς
      min <-- R[1]-F[1]+1
      θmin <-- 1
      για i από 2 μέχρι 5
        Αν F[i] = 0 τότε
          min <-- 0
          θmin <-- i
        ΑλλιώςΑν R[i]-F[i]+1 < min τότε
          min <-- R[i]-F[i]+1
          θmin <-- i
        ΤέλοςΑν
      ΤέλοςΕπανάληψης
    ΤέλοςΑν
    Αν min = 100 τότε
      Γράψε 'Γεμάτες ουρές'
    Αλλιώς
      Αν F[θmin] = 0 τότε
        F[θmin] <-- 1
        R[θmin] <-- 1
      Αλλιώς
        R[θmin] <-- R[θmin] + 1
      ΤέλοςΑν
      Διάβασε Q[θmin, R[θmin]]
    ΤέλοςΑν
  ΑλλιώςΑν επ = 2 τότε
    ΑρχήΕπανάληψης
      Διάβασε x
    ΜέχριςΌτου x >= 1 ΚΑΙ x <= 5
    Αν F[x] = 0 τότε
      Γράψε 'Άδεια ουρά'
    Αλλιώς
      Γράψε Q[x, F[x]]
      Αν F[x] = R[x] τότε
        F[x] <-- 0
        R[x] <-- 0
      Αλλιώς
        F[x] <-- F[x] + 1
      ΤέλοςΑν
      A[x] <-- A[x] + 1
    ΤέλοςΑν
  ΤέλοςΑν
ΜέχριςΌτου επ = 3
! max με ισοτιμία στον A[5]...
```