Μονοδιάστατοι Πίνακες

Τι είναι ο πίνακας γενικά :

Πίνακας είναι μια **Στατική Δομή Δεδομένων**. Δηλαδή συνεχόμενες θέσεις μνήμης, όπου το πλήθος των θέσεων είναι συγκεκριμένο. Στις θέσεις αυτές καταχωρούμε – αποθηκεύουμε τιμές μεταβλητών - δεδομένων ώστε να παραμένουν στη μνήμη για περαιτέρω επεξεργασία και διαχείριση.

Πιο αναλυτικά θα μπορούσε να δοθεί σαν ορισμός το παρακάτω :

Πίνακας είναι ένα σύνολο από καθορισμένο πλήθος θέσεων μνήμης, που κάθε μια δέχεται μια τιμή (όπως οι κοινές μεταβλητές), προσδιορίζονται όλες ενιαία από ένα κοινό όνομα, τον ίδιο τύπο δεδομένων (Ακέραιοι, Πραγματικοί, Χαρακτήρες, Λογικοί) και ένα δείκτη που υποδεικνύει την θέση της κάθε τιμής στον πίνακα.

Συμβολισμός :

Στις εκφωνήσεις ένας πίνακας π.χ. 10 θέσεων συμβολίζεται Α(10).

Στη δήλωση μεταβλητών ενός προγράμματος συμβολίζεται Α[1..10].

Στο λογισμικό 'Διερμηνευτή Γλώσσας' στη δήλωση μεταβλητών συμβολίζεται Α[10]

Αριθμός Θέσης

Έστω ένας πίνακας με όνομα Α δέκα θέσεων :

$1^η$	$2^η$	$3^η$	$4^η$	$5^η$	$6^η$	$7^η$	$8^η$	$9^η$	$10^η$
5	**7**	**20**	**-15**	**4**	**8**	**-9**	**-21**	**20**	**7**

Οπότε αν θέλουμε το περιεχόμενο – τη τιμή κάποιας θέσης αρκεί να γράψουμε το όνομα του πίνακα με τον αριθμό της θέσης.

Π.χ. Α[1]=5, Α[5]=4, Α[8]=-21, α[9]=20, Α[10]=7

Γενικά : Α [Ι]

Περιεχόμενο - τιμή

Όνομα

Δείκτης - Θέση

Για να γεμίζουμε – διαβάσουμε, για να εμφανίσουμε και επεξεργαστούμε τιμές σε ένα πίνακα πρέπει να χρησιμοποιήσουμε μια επαναληπτική δομή. Επειδή είναι πάντα γνωστός ο αριθμός των θέσεων (κόμβων) – στοιχείων του, συνήθως επιλέγουμε την δομή **ΓΙΑ**. Ωστόσο η όποια διαδικασία υλοποιείται και τις άλλες δύο δομές επανάληψης ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ.

Σε κάθε δομή δεδομένων και άρα σε έναν πίνακα τα στοιχεία υφίστανται επεξεργασία από ένα σύνολο λειτουργιών – πράξεων, όπου οι βασικές είναι εξής :

1. **Προσπέλαση :** πρόσβαση στα δεδομένα με σκοπό την οποιαδήποτε επεξεργασία τους (ανάγνωση, εγγραφή, μετακίνηση, αντικατάσταση κ.α.)
2. **Αναζήτηση (έχουμε 2 τύπους αναζήτησης)**
3. **Μεταβολή :** η αλλαγή του περιεχομένου ενός κόμβου.
4. **Ανάκτηση :** η λήψη του περιεχόμενου ενός κόμβου με οποιονδήποτε τρόπο.
5. **Ταξινόμηση:** η διάταξη των στοιχείων μιας δομής δεδομένων σε αύξουσα ή φθίνουσα σειρά.
6. **Εισαγωγή :** η προσθήκη ή δημιουργία νέων κόμβων σε μια υπάρχουσα δομή.
7. **Διαγραφή :** το αντίθετο της Εισαγωγής

Οι πέντε πρώτες εφαρμόζονται και στους δύο τύπους δομών δεδομένων, δηλαδή στις Στατικές και Δυναμικές.

Οι δύο τελευταίες εφαρμόζονται μόνο στις Δυναμικές δομές (Στοίβα και Ουρά).
Από τις πέντε πρώτες θα αναλύσουμε κάθε μια λειτουργία - πράξη για τους πίνακες με τη άνωθεν σειρά. Επί της ουσίας δίνονται οι αλγόριθμοι που εκφράζουν τις λειτουργίες αυτές. Οπότε αν κάποιος καταλάβει και γνωρίζει τις βασικές αυτές λειτουργίες μπορεί να τις συνδυάσει και να λύσει τα προβλήματα που επιλύονται με τη χρήση πινάκων.
Πρώτα όμως **πρέπει να απαντήσουμε στο ερώτημα : Γιατί και πότε πρέπει να χρησιμοποιήσουμε πίνακα ; Η απάντηση είναι απλή : Όταν για να λυθεί το πρόβλημα απαιτείται η παραμονή κάποιων δεδομένων στη μνήμη του υπολογιστή. Αυτό γιατί, επειδή ο πίνακας δεσμεύει ένα κομμάτι μνήμης για τα δεδομένα, τα 'κρατά' καθ' όλη τη διάρκεια εκτέλεσης του αλγορίθμου – προγράμματος, δίνοντάς μας έτσι τη δυνατότητα να τα επεξεργαστούμε – διαχειριστούμε όταν και όσες φορές θέλουμε.**
Από το γέμισμα ενός πίνακα και την απόδοση τιμών σε μια μεταβλητή τόσες φορές όσες είναι οι θέσεις (κόμβοι) του πίνακα, η παραπάνω έννοια γίνεται αμέσως κατανοητή και επιβεβαιώνεται από την εμφάνιση των στοιχείων.

Για Ι από 1 μέχρι 10
  Διάβασε Α[Ι]
Τέλος_Επανάληψης
Για Ι από 1 μέχρι 10
  Γράψε Α[Ι]
Τέλος_Επανάληψης
	Για Ι από 1 μέχρι 10
	  Διάβασε Α
	Τέλος_Επανάληψης
	  Γράψε Α

Όλες οι τιμές – δεδομένα παρέμειναν στη μνήμη, αποθηκευμένες στις αντίστοιχες θέσεις και εμφανίζονται όλες.	Καμία τιμή – δεδομένο δεν παρέμεινε στη μνήμη, εκτός του τελευταίου που δόθηκε, οπότε μόνο αυτό εμφανίζεται.

ΠΡΟΣΠΕΛΑΣΗ

Η προσπέλαση είναι η πράξη με την οποία μπορούμε να μεταβούμε σε κάποιο στοιχείο (κόμβο) του πίνακα με ή χωρίς να σαρώσουμε τα προηγούμενα στοιχεία (κόμβους) του. Ολική Σάρωση έχουμε όταν προσπελαύνουμε ένα – ένα (από την αρχή μέχρι το τέλος) όλα τα στοιχεία του πίνακα.
Για τον πίνα Α(10) του προηγούμενου παραδείγματος θα μπορούσαμε να κάνουμε τις παρακάτω προσπελάσεις (χωρίς σάρωση) :
Α[6] ← Α[6] + 7
Α[2] ← Α[1]
Α[7] ← 5
Α[3] ← Α[4]+(Α[9] DIV 2)
Α[9] ← 2*Α[5] – Α[8]
Οπότε ο πίνακας Α(10) θα έχει αλλάξει ως εξής :

5	**5**	**-3**	**-15**	**4**	**15**	**5**	**-21**	**29**	**7**

Παρατηρούμε ότι : α) άλλαξαν μόνο οι τιμές των θέσεων όπου μεταβήκαμε και κάναμε κάποιες πράξεις, β) τα νέα στοιχεία – δεδομένα του πίνακα Α συνεχίζουν να παραμένουν στη μνήμη για οποιαδήποτε περαιτέρω επεξεργασία, γ) κάθε θέση του πίνακα συμπεριφέρεται ακριβώς όπως και μια μεταβλητή, χωρίς καμία απολύτως διαφορά, δ) οι θέσεις του πίνακα μπορούν να λειτουργήσουν και σαν μεμονωμένες μεταβλητές και ενιαία ως θέσεις του πίνακα.

ΑΝΑΖΗΤΗΣΗ

Η αναζήτηση δεν τίποτα άλλο από τη πράξη που κάνουμε για να βρούμε κάτι συγκεκριμένο σε ένα πίνακα. Βασίζεται στη πράξη της προσπέλασης και μας δίνει τη δυνατότητα να αναζητήσουμε στοιχεία βάσει του περιεχόμενου των θέσεων, βάσει του

αύξοντα αριθμού των θέσεων και συνδυάζοντας και τα δύο προηγούμενα. Η αναζήτηση γίνεται κυρίως για να επεξεργαστούμε, να διαχειριστούμε, να επιβεβαιώσουμε κάποιο ή κάποια στοιχεία ενός πίνακα.
Υπάρχουν δύο αλγόριθμοι αναζήτησης, που για ευκολία θα του λέμε Αναζήτηση Α και Αναζήτηση Β.

<u>Αναζήτηση Α</u>
Την εφαρμόζουμε όταν πρέπει να σαρώσουμε όλα τα στοιχεία (κόμβους) ενός πίνακα. Έστω ένας πίνακα Α(20) με ακεραίους.

Επεξεργασία στοιχείων πίνακα βάσει περιεχομένου. Π.χ. Το άθροισμα των αρτίων αριθμών	**Επεξεργασία στοιχείων πίνακα βάσει της θέσης.** Π.χ. Το γινόμενο των αριθμών που βρίσκονται στις άρτιες θέσεις.	**Συνδυασμός και των δύο.** Π.χ. Τον μέσο όρο των αριθμών που βρίσκονται σε περιττές θέσεις αλλά είναι άρτοι αριθμοί.
Σ ← 0 Για Ι από 1 μέχρι 20 Αν Α[Ι] MOD 2 = 0 τότε Σ ← Σ + Α[Ι] Τέλος_Αν Τέλος_Επανάληψης Γράψε Σ	Ρ ← 0 Για Ι από 1 μέχρι 20 Αν Ι MOD 2 = 0 τότε Ρ ← Ρ + Α[Ι] Τέλος_Αν Τέλος_Επανάληψης Γράψε Ρ	Σ ← 0, Μ ← 0 Για Ι από1 μέχρι 20 Αν (Ι MOD 2 = 1) και (Α[Ι] MOD 2 =0) τότε Σ ← Σ + Α[Ι] Μ ← Μ+1 Τέλος_Αν Τέλος_Επανάληψης ΜΟ ← Σ/Μ Γράψε ΜΟ

Διαπιστώνουμε ότι και στις τρεις περιπτώσεις σαρώσαμε όλα τα στοιχεία του πίνακα ακόμη κι όταν βρίσκαμε τα στοιχεία που θέλαμε.
Άσκηση : Να δοθούν 20 ακέραιοι αριθμοί και να βρεθεί το πλήθος των αριθμών με την μέγιστη τιμή.
Κατ' αρχήν θα πρέπει να απαντηθεί η ερώτηση αν πρέπει ή όχι να χρησιμοποιήσουμε πίνακα για τους 20 αυτούς αριθμούς. Η απάντηση είναι ΝΑΙ γιατί, ενώ για να βρούμε τη μέγιστη τιμή δεν χρειάζεται πίνακας, για να βρούμε όμως το πλήθος τους θα ήταν καλύτερα οι τιμές των αριθμών να παραμείνουν στη μνήμη -- ακολουθεί αναλυτική επεξήγηση. (Το πρόβλημα ωστόσο λύνεται και χωρίς τη χρήση πίνακα, αλλά είναι ένα καλό παράδειγμα για την κατανόηση της έννοιας του πίνακα).
Αρχικά θα πρέπει να βρούμε πρώτα την μέγιστη τιμή και μετά το πλήθος. Θεωρούμε την τιμή της πρώτης θέσης του πίνακα ως τη μεγαλύτερη και την καταχωρούμε σε μια μεταβλητή ΜΑΧ. Στη συνέχεια σαρώνουμε τον πίνακα και όποιος αριθμός είναι μεγαλύτερος αντικαθιστά τη τιμή του ΜΑΧ.
Μετά ξανασαρώνουμε το πίνακα και ελέγχουμε- αναζητάμε ποια στοιχεία έχουν την ΜΑΧ τιμή και πράττουμε ανάλογα.

```
Αρχή
ΜΑΧ ← Α[1]
Για Ι από 2 μέχρι 20
 Αν Α[Ι] > ΜΑΧ τότε
   ΜΑΧ ← Α[Ι]
 Τέλος_Αν
Τέλος_Επανάληψης
  Γράψε ΜΑΧ
ΠΛ ← 0
Για Ι από 1 μέχρι 20
```

Αν Α[Ι] = ΜΑΧ τότε
ΠΛ ← ΠΛ + 1
Τέλος_Αν
Τέλος_Επανάληψης
Γράψε ΠΛ
Τέλος

<u>Αναζήτηση Β</u>
Την εφαρμόζουμε όπου όταν βρεθεί το αναζητούμενο στοιχείο τερματίζει ο αλγόριθμος. Δηλαδή, αυτό που ψάχνουμε να είναι μοναδικό ή για κάποιο λόγο δεν χρειάζεται να συνεχιστεί η αναζήτηση.

Έστω ότι ψάχνουμε στο πίνακα Α(20) με ακεραίους έναν αριθμό τον οποίο θα δίνουμε από το πληκτρολόγιο.
Αρχή
Διάβασε Ζ
Βρέθηκε ← Ψευδής
Θ ← 0
Ι ← 1
Όσο Βρέθηκε = Ψευδής και Ι <= 20 επανάλαβε
Αν Α[Ι] = Ζ τότε
Βρέθηκε ← Αληθής
Θ ← Ι
Αλλιώς
Ι ← Ι + 1
Τέλος_Αν
Τέλος_Επανάληψης
Τέλος

- Χρησιμοποιώ την δομή επανάληψης ΟΣΟ γιατί δεν γνωρίζω εξ' αρχής σε ποια φορά της επανάληψης θα βρω το στοιχείο.
- Χρησιμοποιώ και μια λογική μεταβλητή, η οποία έχει αρχική τιμή Ψευδής και **όταν βρω το ζητούμενο παίρνει τη τιμή Αληθής και τερματίζει ο αλγόριθμος**.
- Αν το ζητούμενο βρίσκεται στη τελευταία θέση του πίνακα, τότε θα σαρωθεί ολόκληρος ο πίνακας.
- Αν η μεταβλητή *Βρέθηκε* παραμείνει με την τιμή *Ψευδής* σημαίνει ότι το ζητούμενο δεν υπάρχει στον πίνακα Α, ενώ αν πάρει την τιμή *Αληθής* ότι ζητούμενο υπάρχει στον πίνακα.

ΜΕΤΑΒΟΛΗ - ΤΡΟΠΟΠΟΙΗΣΗ
Η λειτουργία της μεταβολής έχει σκοπό την αλλαγή – τροποποίηση του περιεχομένου ενός ή πολλών κόμβων σε μια δομή (πίνακα), βάση κάποιου ή κάποιων κανόνα/νων.
Έστω ο πίνακας Α(10) με τιμές :
Α

5	2	-4	6	-5	-7	8	12	-2	9

Και ένα πίνακας Β με τις ίδιες τιμές :
Β

5	2	-4	6	-5	-7	8	12	-2	9

Παράδειγμα 1	**Παράδειγμα 2**

Έστω ο πίνακας A(10) με ακεραίους. Οι αριθμοί που είναι πολλαπλάσια του 3 να διπλασιαστούν, και ο νέος αριθμός να αυξηθεί κατά μια μονάδα.	Έστω ο πίνακας B(10) με ακεραίους. Οι αριθμοί που βρίσκονται στις άρτιες θέσεις να αντικατασταθούν με το 1 και οι αριθμοί που βρίσκονται στις περιττές θέσεις να αντικατασταθούν με το 2.
Για I από 1 μέχρι 10 Αν A[I] MOD 3 = 0 τότε A[I] ← 2*A[I] + 1 Τέλος_Αν Τέλος_Επανάληψης Για I από 1 μέχρι 10 Γράψε A[I] Τέλος_Επανάληψης	Για I από 1 μέχρι 10 Αν I MOD 2 = 0 τότε B[I] ← 1 Αλλιώς B[I] ← 2 Τέλος_Αν Τέλος_Επανάληψης Για I από 1 μέχρι 10 Γράψε B[I] Τέλος_Επανάληψης

Οπότε ο πίνακας A θα έχει τιμές :

5	2	-4	**13**	-5	-7	8	**25**	-2	**28**

και ο πίνακας B θα έχει τιμές :

2	1	2	1	2	1	2	1	2	1

Ουσιαστικά η πράξη αυτή βασίζεται στις 2 προηγούμενες (προσπέλαση και αναζήτηση).

ΤΑΞΙΝΟΜΗΣΗ

Η Ταξινόμηση είναι η σημαντικότερη λειτουργία – πράξη και έχει να κάνει με τη διάταξη των στοιχείων - τιμών ενός πίνακα σε αύξουσα ή φθίνουσα σειρά. Χρησιμοποιείται για την επίλυση πολλών προβλημάτων, διότι η εφαρμογή της διευκολύνει στην εύρεση λύσεων πολλών περιπτώσεων.

Ο αλγόριθμος που προτείνει το βιβλίο σας είναι με τη μέθοδο της ΕΠΙΛΟΓΗΣ και βασίζεται στα παρακάτω βήματα :

1) επιλογή του στοιχείου με την ελάχιστη τιμή, 2) αντιμετάθεση των τιμών τους με το προηγούμενο στοιχείο, 3) επανάληψη των βημάτων 1) και 2) με τα υπόλοιπα στοιχεία.

```
Για I από 1 μέχρι 10
        J ← I
        Min ← A[J]
    Για K από I+1 μέχρι 10
        Αν A[K] < Min τότε
                J ← K
                Min ← A[J]
        Τέλος_Αν
    Τέλος_επανάληψης
    Temp ← A[I]
     A[I] ← A[J]
     A[J] V Temp
Τέλος_επανάληψης
```

Υπάρχει όμως και ο αλγόριθμος της ταξινόμησης βασίζεται στη μέθοδο της ΦΥΣΑΛΙΔΑΣ, όπου συγκρίνουμε δύο διαδοχικά στοιχεία (περιεχόμενα κόμβων) του πίνακα και τα αντιμεταθέτουμε ανάλογα αν το ένα είναι μεγαλύτερο ή μικρότερο από άλλο (για ορίσουμε την αύξουσα ή φθίνουσα σειρά). Η διαδικασία αυτή (που γίνεται για όλα τα στοιχεία, δηλ. είναι σάρωση) γίνεται τόσες φορές όσο είναι το πλήθος των στοιχείων(κόμβων) του πίνακα μείον ένα. Σε κάθε σάρωση μετακινείται πάντα προς τα πάνω το μικρότερο σε τιμή στοιχείο. Άρα για ένα πίνακα Α(10) έχουμε :

```
Για Ι από 2 μέχρι 10
  Για J από 10 μέχρι Ι με βήμα -1
    Αν A[J-1] > A[J] τότε
        X ← A[J-1]
        A[J-1] ← A[J]
        A[J] ← X
    Τέλος_Αν
  Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

Ο παραπάνω αλγόριθμος ταξινομεί σε αύξουσα σειρά, αν αλλάξουμε την ανισότητα στη δομή Αν τότε η ταξινόμηση θα γίνει σε φθίνουσα σειρά.

Η εξωτερική επανάληψη αντιστοιχεί στις ΣΑΡΩΣΕΙΣ, η εσωτερική επανάληψη στις ΣΥΓΚΡΙΣΕΙΣ και η δομή Αν στις ΑΝΤΙΜΕΤΑΘΕΣΕΙΣ.

Πλεονέκτημα Αλγορίθμου: είναι σχετικά εύκολος στην υλοποίησή του και θα ταξινομήσει τον πίνακα με οποιαδήποτε διάταξη στοιχείων έχει.

Μειονέκτημα Αλγορίθμου : δεν είναι τόσο ‘έξυπνος’, γιατί στην περίπτωση που ο πίνακας είναι ήδη ταξινομημένος ή ταξινομηθεί σε λιγότερες σαρώσεις(επαναλήψεις) που είναι πολύ πιθανό, ο αλγόριθμος συνεχίζει να κάνει όλες τις εξωτερικές και εσωτερικές επαναλήψεις χωρίς κανένα επί της ουσίας αποτέλεσμα.

Η Άσκηση ΔΤ2 σελ. 33 από τετράδιο μαθητή, εστιάζει στο μειονέκτημα αυτό και σας προτείνει να βελτιώσετε τον αλγόριθμο αυτό κάνοντάς πιο’ έξυπνο’. Δηλαδή να τον τροποποιήσετε έτσι ώστε όταν διαπιστωθεί ότι ο πίνακας ταξινομήθηκε να τερματίζει, χωρίς να κάνει άλλες περιττές σαρώσεις(επαναλήψεις).

ΛΥΣΗ

- Όταν σε κάποια σάρωση δεν υπάρξει αντιμετάθεση τότε ο πίνακας είναι ταξινομημένος.
- Αν π.χ. ο πίνακας ταξινομηθεί στη $4^η$ σάρωση, θα κάνει άλλη μια (δηλ. την τη $5^η$) και μετά τα τερματίσει. Αυτό γιατί στην $5^η$ σάρωση θα διαπιστωθεί ότι δεν υπήρξε αντιμετάθεση.
- Θα χρησιμοποιήσουμε μια λογική μεταβλητή για το σκοπό αυτό.
- Η εξωτερική επανάληψη θα αντικατασταθεί με τη δομή ΜΕΧΡΙΣ_ΟΤΟΥ, γιατί δεν γνωρίζουμε εκ των προτέρων το πλήθος των σαρώσεων (δεν ξέρουμε σε ποια σάρωση δεν θα υπάρξει αντιμετάθεση – ενώ ο πίνακας θα έχει ταξινομηθεί στην αμέσως προηγούμενη σάρωση).
- Θεωρούμε πάντα ότι πριν από νέα σάρωση ο πίνακας είναι ταξινομημένος.

```
Αρχή
   I ← 2
   Αρχή_Επανάληψης
     FLAG ← Ψευδής
       Για J από 10 μέχρι Ι με βήμα -1
              Αν A[J-1] > A[J] τότε
                X ← A[J-1]
                A[J-1] ← A[J]
                A[J] ← X
```

```
                FLAG ← Αληθής
            Τέλος_Αν
        Τέλος_Επανάληψης
        I ← I + 1
    Μέχρις_ότου  (FLAG = Ψευδής) ή  (I >= 10)
Τέλος
```

Πως θα απαντήσετε στα ερωτήματα : Πως θα εμφανιστούν οι πέντε μεγαλύτεροι βαθμοί μιας τάξης 30 μαθητών ? Ποιος είναι ο μεγαλύτερος βαθμός της τάξης ? Πόσοι έχουν το μεγαλύτερο βαθμό ? Πως θα εμφανίζατε και τα αντίστοιχα ονόματά τους για τα δύο πρώτα ερωτήματα ?

ΑΝΑΚΤΗΣΗ

Βασική προϋπόθεση για την πραγματοποίηση των προηγούμενων λειτουργιών σε μια δομή (π.χ. Πίνακα) είναι η ανάκτηση του περιεχομένου ενός κόμβου. Δηλαδή να είναι δυνατή η λήψη (ανάγνωση) του στοιχείου αυτού, καθώς και των υπολοίπων.