Έστω ένας πίνακας με όνομα Α δέκα θέσεων (π.χ. με τις παρακάτω τιμές) :

$1^η$	$2^η$	$3^η$	$4^η$	$5^η$	$6^η$	$7^η$	$8^η$	$9^η$	$10^η$
5	**7**	**20**	**-13**	**4**	**8**	**-9**	**-21**	**20**	**7**

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) :

A[6] ← A[6] + 7
A[2] ← A[1]
A[7] ← 5
A[3] ← A[4]+(A[9] DIV 2)
A[9] ← 2*A[5] – A[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 A[I] MOD 2 = 0 τότε Σ ← Σ + A[I] Τέλος_Αν Τέλος_Επανάληψης Γράψε Σ	Ρ ← 0 Για Ι από 1 μέχρι 20 Αν I MOD 2 = 0 τότε Ρ ← Ρ + A[I] Τέλος_Αν Τέλος_Επανάληψης Γράψε Ρ	Σ ← 0, Μ ← 0 Για Ι από1 μέχρι 20 Αν (I MOD 2 = 1) και (A[I] MOD 2 =0) τότε Σ ← Σ + A[I] 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.

ΑΝΤΙΓΡΑΦΗ

Η αντιγραφή είναι όπως το λέει και η ίδια η λέξη, δηλαδή να αντιγράφουμε στοιχεία από έναν πίνακα σε έναν άλλο, αλλά στοιχεία από πολλούς πίνακες σε κάποιον άλλο. Βασίζεται στην προσπέλαση και στην αναζήτηση (όταν θέλουμε να αντιγράψουμε μερικά και όχι όλα τα στοιχεία ενός πίνακα). Έχουμε όμως δύο περιπτώσεις όπου η αντιγραφή δεν βασίζεται στον ίδιο αλγόριθμο.

Έστω ο πίνακας Α(10) με τιμές :

Α

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

Όταν οι πίνακες έχουν την ίδια διάσταση, δεν υπάρχει κανένα πρόβλημα στην αντιγραφή στοιχείων από τον έναν στον άλλο.	Όταν όμως πρέπει να αντιγραφούν στοιχεία από έναν πίνακα σε άλλον με διαφορετική διάσταση, τότε πρέπει να κάνουμε κάποιες επιπλέον ενέργειες και να αλλάξουμε τον αλγόριθμο της αντιγραφής.
Έστω ένας πίνακας Α(10) με ακεραίους. Οι αριθμοί που είναι πολλαπλάσια του 3 να διπλασιαστούν, ο νέος αριθμός να αυξηθεί κατά μια μονάδα και να αντιγραφούν όλα τα στοιχεία του πίνακα Α(10) στον Β(10). Ο πίνακας Β έχει κι αυτός 10 θέσεις οπότε κάθε αριθμός θα αντιγραφή στην αντίστοιχή του από τον πίνακα Α.	Έστω ένας πίνακας Α(10) με ακεραίους. Να αντιγραφούν σε έναν άλλο πίνακα Β(?) οι αρνητικοί αριθμοί.. Για τον πίνακας Β δεν μπορούμε να γνωρίζουμε την διάστασή του, γιατί δεν γνωρίζουμε το πλήθος των αρνητικών, που μπορεί να είναι από 0 μέχρι και 10. Οπότε ορίζουμε τον Β με 10 θέσεις και γεμίζει ανάλογα με το πλήθος των ζητούμενων στοιχείων. Όσο για την αναντιστοιχία των θέσεων (π.χ. εκείνος που είναι στην $3^η$ θέση του Α πρέπει να καταχωρηθεί στην $1^η$ του Α, εκείνος της $5^{ης}$ θέσης του Α στην $2^η$ του Β κ.ο.κ.) λύνεται με την παρακάτω τεχνική.
Για Ι από 1 μέχρι 10 Αν Α[Ι] MOD 3 = 0 τότε Α[Ι] ← 2*Α[Ι] + 1 Τέλος_Αν Τέλος_Επανάληψης Για Ι από 1 μέχρι 10 Β[Ι] ← Α[Ι] Τέλος_Επανάληψης Για Ι από 1 μέχρι 10 Γράψε Β[Ι] Τέλος_Επανάληψης	Κ ← 0 Για Ι από 1 μέχρι 10 Αν Α[Ι] < 0 τότε Κ ← Κ + 1 Β[Κ] ← Α[Ι] Τέλος_Αν Τέλος_Επανάληψης Για Ι από 1 μέχρι Κ Γράψε Β[Ι] Τέλος_Επανάληψης

Οπότε ο πίνακας Β για κάθε περίπτωση θα έχει τιμές :

1) Β

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

2) Β

-4	-3	-7	-2						

Με τη λογική αυτή της 2ης περίπτωσης μπορούμε να αντιγράφουμε στοιχεία του Α σε άλλο πίνακα, αλλά όχι απαραίτητα από την αρχή, κάπου στη μέση και σε ενδιάμεσους κόμβους.

Ο πίνακας Β είναι κι αυτός στατική δομή και δηλώνεται με το μέγιστο δυνατό πλήθος θέσεων, όμως γεμίζει σαν δυναμική δομή και είναι πολύ πιθανό ένα τμήμα του να μην πάρει ποτέ τιμές, ενώ η μνήμη αυτή έχει δεσμευτεί.

ΣΥΓΧΩΝΕΥΣΗ

Η Συγχώνευση είναι η συνένωση δύο ή περισσοτέρων πινάκων σε έναν νέο πίνακα που αποτελείται από τα στοιχεία των αρχικών πινάκων. Σημασία όμως έχει η σειρά καταχώρησης των εμπλεκόμενων πινάκων. Ο νέος πίνακας θα έχει διάσταση ίση με το άθροισμα των διαστάσεων των αρχικών πινάκων. Βασίζεται στη προσπέλαση, την αντιγραφή και ίσως(κάποιες φορές) την αναζήτηση.

Έστω δύο πίνακες Α(5) και Β(9). Να δημιουργηθεί ένας νέος Γ(14) που θα περιέχει τα στοιχεία του Α και Β με την αντίστοιχη σειρά.

Α(5)

5	**-2**	**-3**	**4**	**6**

Β(9)

9	**5**	**-2**	**7**	**-7**	**6**	**4**	**3**	**7**

Γ(14)

5	**-2**	**-3**	**4**	**6**	**9**	**5**	**-2**	**7**	**-7**	**6**	**4**	**3**	**7**

Για Ι από 1 μέχρι 5
  Γ[Ι] ← Α[Ι]
Τέλος_Επανάληψης
Για Ι από 1 μέχρι 9
  Γ[Ι+5] ← Β[Ι]
Τέλος_Επανάληψης

ΔΙΑΧΩΡΙΣΜΟΣ

Ο Διαχωρισμός πινάκων είναι ακριβώς η αντίθετη διαδικασία. Δηλαδή είναι η δημιουργία δύο ή περισσοτέρων πινάκων από τα στοιχεία ενός πίνακα. Βασίζεται στη προσπέλαση, την αντιγραφή και την αναζήτηση. Έχουμε δύο περιπτώσεις διαχωρισμού πινάκων : 1) όταν διαχωρίζουμε βάσει των θέσεων του πίνακα και 2) όταν διαχωρίζουμε βάσει του περιεχομένου του πίνακα.

Έστω ο πίνακας Α(10) με στοιχεία :

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

1) Να χωριστεί σε δύο νέους πίνακες Β, Γ όπου ένας να αποτελείται από τα 4 πρώτα στοιχεία(κόμβους) και ο άλλος από τα υπόλοιπα 6 στοιχεία(κόμβους).

Για Ι από 1 μέχρι 4
  Β[Ι] ← Α[Ι]
Τέλος_Επανάληψης
Για Ι από 1 μέχρι 6
  Γ[Ι] ← Α[Ι+4]
Τέλος_Επανάληψης

Ο πίνακας Β θα είναι :

5	-2	-3	4

Ο πίνακας Γ θα είναι :

7	4	2	6	-8	1

2) Να χωριστεί σε δύο νέους πίνακες Β, Γ όπου ο ένας θα περιέχει τους θετικούς και ο άλλος τους αρνητικούς αριθμούς.

Το πρόβλημά μας εδώ είναι δεν γνωρίζουμε το πλήθος των θετικών και των αρνητικών αντίστοιχα, ώστε να ορίσουμε τις διαστάσεις των νέων πινάκων ως στατικές δομές που πρέπει να είναι. Το πρόβλημα αντιμετωπίζεται με τη μέθοδο που εφαρμόσαμε στη πράξη της Αντιγραφής. Οπότε :

```
  Κ ← 0
  Χ ← 0
Για Ι από 1 μέχρι 10
  Αν Α[Ι] > 0 τότε
    Κ ← Κ + 1
    Β[Κ] ← Α[Ι]
  Αλλιώς
  Γ[Χ] ← Α[Ι]
  Τέλος_Αν
Τέλος_Επανάληψης
```

Ο πίνακας Β θα είναι :

5	4	7	4	2	6	1			

Ο πίνακας Γ θα είναι :

-2	-3	-8							

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 της ταξινόμησης βασίζεται στη μέθοδο της φυσαλίδας, 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 Α[J-1] > Α[J] τότε
        Χ ← Α[J-1]
        Α[J-1] ← Α[J]
        Α[J] ← Χ
    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 I 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)
```

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 ?