ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ - ΑΛΓΟΡΙΘΜΟΙ ΑΝΑΖΗΤΗΣΗΣ/ΤΑΞΙΝΟΜΗΣΗΣ

- Τι γνωρίζετε για τις δομές δεδομένων δευτερεύουσας μνήμης;

- Στα αρχεία τα δεδομένα δεν χάνονται όταν σβήσει ο υπολογιστής, ενώ στις δομές δεδομένων κύριας μνήμης χάνονται με το τέλος του αγλορίθμου
- Τα αρχεία αποτελούνται από εγγραφές. Κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία που περιγράφουν τα χαρακτηριστικά της εγγραφής π.χ. Μητρώο, ονοματεπώνυμο
- Ένα πεδίο ταυτοποιεί (προσδιορίζει μοναδικά) την εγγραφή και ονομάζεται πρωτεύον κλειδί ή απλά κλειδί (π.χ. μητρώο μαθητή). Κάποια άλλα πεδία μπορεί να χρησιμοποιηθούν ως δευτερεύοντα κλειδιά (π.χ. επώνυμο μαθητή)
- Η αναζήτηση μια εγγραφής με βάση την τιμή του πρωτεύοντος ή δευτερεύοντος κλειδιού αποτελεί ερευνητικό θέμα αρκετά σημαντικό και ενδιαφέρον
- Όταν ο πίνακας δεν είναι ταξινομημένος χρησιμοποιούμε μέθοδο σειριακής αναζήτησης, όταν ο πίνακας είναι ταξινομημένος χρησιμοποιούμε δυαδική μέθοδο αναζήτησης.

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΣΕΙΡΙΑΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Έστω ότι έχουμε τον παρακάτω μονοδιάστατο πίνακα και θέλουμε να βρούμε αν υπάρχει ή όχι το στοιχείο 4 στον πίνακα. Θα ελέγξουμε ένα προς ένα τα στοιχεία του πίνακα κάνοντας έλεγχο αν ισούνται με 4.

14	3	-2	-2	4	2

Αν βρούμε το στοιχείο που ψάχνουμε τότε εμφανίζουμε τη θέση του και ενημερώνουμε το χρήστη. Αν ο πίνακας ελεγχθεί όλος χωρίς να βρούμε το στοιχείο, τότε εμφανίζουμε ένα μήνυμα στον χρήστη ότι το στοιχείο δεν βρέθηκε.

- Ο έλεγχος των στοιχείων του πίνακα γίνεται με μια επανάληψη και ένα Αν… τότε. π.χ. Αν table[i] = key τότε, όπου key το στοιχείο αναζήτησης. Αν η συνθήκη ισχύει τότε εμφανίζουμε τη θέση i που βρέθηκε το στοιχείο.
- Για να μπορέσουμε να εμφανίσουμε ένα μήνυμα σχετικά με την μη ύπαρξη του στοιχείου αναζήτησης, χρησιμοποιούμε μια βοηθητική λογική μεταβλητή (με όνομα βρέθηκε ή flag ή found, κ.ο.κ), στην οποία θέτουμε την τιμή Ψευδής πριν την επανάληψη. Αν κατά τον έλεγχο βρεθεί το στοιχείο που αναζητούμε, τότε εκχωρούμε στη λογική μεταβλητή την τιμή Αληθής. Έτσι αν στο τέλος της επανάληψης η λογική μεταβλητή είναι Αληθής, σημαίνει ότι βρέθηκε η τιμή που αναζητούμε στον πίνακα. Αν η λογική μεταβλητή είναι Ψευδής, η τιμή που αναζητούμε δεν βρέθηκε. Το μήνυμα για την μη ύπαρξη πρέπει να δοθεί στο τέλος της επανάληψης

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΣΕΙΡΙΑΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Αλγόριθμος Σειριακή_Αναζήτηση ! **Εμφάνιση 1ης θέσης**
Δεδομένα // N, Table, key//
! *N: διάσταση πίνακα, Table: πίνακας, key: τιμή αναζήτησης*

```
flag ← Ψευδής
i ← 1
pos ← -1
Όσο i <= N και flag = Ψευδής επανάλαβε
      Αν Table[i] = key τότε
          Εμφάνισε “Η τιμή βρέθηκε στη θέση “, i
          pos ← i
          flag ← Αληθής
      Αλλιώς
          i <- i + 1
      Τέλος_αν
Τέλος_επανάληψης
Αν flag ← Ψευδής τότε
    Εμφάνισε “Η τιμή δεν βρέθηκε στον πίνακα”
Τέλος_αν
Τέλος Σειριακή_Αναζήτηση
```

! Αρχικοποιώ την μεταβλητή θέσης που βρέθηκε το στοιχείο pos με κάποια τιμή που δεν μπορεί να είναι αληθής π.χ. -1. Μπορούσα να βάλω οποιαδήποτε άλλη τιμή.
! Χρησιμοποιώ την όσο, γιατί δεν ξέρω πόσες επαναλήψεις χρειάζεται να κάνω
! i <= N, γιατί είναι ο μετρητής που “σαρώνει” τα στοιχεία του πίνακα.

! Αν δεν βρω το στοιχείο που ψάχνω πάω στο επόμενο στοιχείο, δηλαδή στο i+1.

! Άρα υπάρχει περίπτωση να μην βρεθεί ποτέ η τιμή στον πίνακα.

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΣΕΙΡΙΑΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

ΠΑΡΑΤΗΡΗΣΕΙΣ

1. Σε περίπτωση που ζητηθεί ο αλγόριθμος σειριακής αναζήτησης ως θεωρητικό θέμα στις τελικές εξετάσεις τότε πρέπει να γραφεί ο αλγόριθμος εμφάνισης της πρώτης θέσης, γιατί αυτόν έχει το σχολικό βιβλίο (*δηλαδή τον προηγούμενο αλγόριθμο*).
2. Ο πίνακας και το στοιχείο αναζήτησης μπορεί να έχουν οποιοδήποτε όνομα, π.χ. Π και x “Αν Π[i] = x τότε”. Η λογική μεταβλητή flag αναφέρεται και με άλλα ονόματα όπως: βρέθηκε, done, stop κτλ.
3. Αν ένα στοιχείο δεν υπάρχει στον πίνακα υποχρεωτικά ελέγχουμε όλες τις τιμές του πίνακα
4. Αν το στοιχείο που ψάχνω υπάρχει παραπάνω από 1 φορά, τότε εμφανίζω τη θέση του 1^{ου} στοιχείου. (*Γι’ αυτό ο αλγόριθμος λέγεται “εμφάνιση 1^{ης} θέσης*)

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΣΕΙΡΙΑΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Αλγόριθμος Σειριακή_Αναζήτηση_Όλες_Οι_Εμφανίσεις

Δεδομένα //N, Table, key//

flag ← **Ψευδής**

Για i **από** 1 **μέχρι** N

Αν Table[i] = key **τότε**

Εμφάνισε “Η τιμή βρέθηκε στη θέση”, i

flag ← **Αληθής**

Τέλος_αν

Τέλος_επανάληψης

! Άρα σαρώνουμε όλο τον πίνακα ασχέτως αν βρω το στοιχείο στην πρώτη θέση. Άρα θα εμφανίσω το στοιχείο σε όλες τις θέσεις που υπάρχει.

! Δεν βάζω Τέλος_αν, γιατί το γράφω στην ίδια ευθεία

Αν flag = **Ψευδής** **τότε** **Εμφάνισε** “Η τιμή δεν βρέθηκε στον πίνακα”

Τέλος Σειριακή_Αναζήτηση_Όλες_Οι_Εμφανίσεις

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΣΕΙΡΙΑΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

- Πότε δικαιολογείται η χρήση της σειριακής μεθόδου αναζήτησης;

Επειδή η σειριακή αναζήτηση είναι αργή διαδικασία, η χρήση της δικαιολογείται στις ακόλουθες περιπτώσεις:

1. Ο πίνακας δεν είναι ταξινομημένος
2. Ο πίνακας είναι μικρού μεγέθους
3. Η αναζήτηση στον πίνακα γίνεται σπάνια

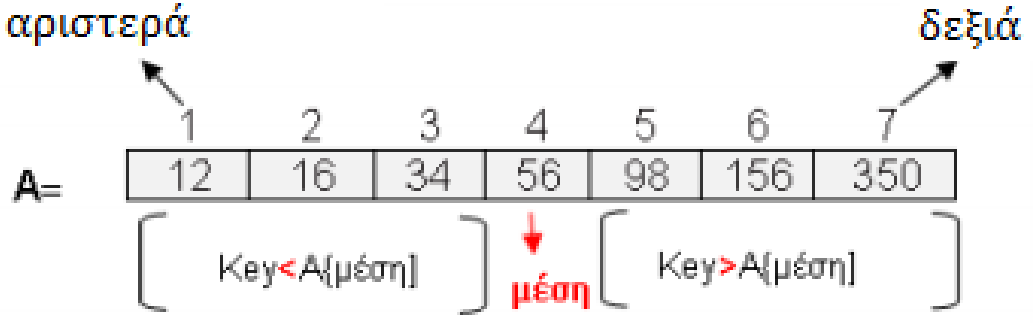
Γ.43. Στη βιβλιοθήκη ενός σχολείου υπάρχουν Ν βιβλία σχετικά με τη γεωγραφία και τα ταξίδια. Έστω ότι κάθε βιβλίο έχει ένα μοναδικό ακέραιο κωδικό και καταχωρείται σε ηλεκτρονικό υπολογιστή ο τίτλος και ο συγγραφέας κάθε βιβλίου. Να γραφεί πρόγραμμα που θα διαβάζει το όνομα ενός συγγραφέα και θα βρίσκει τον κωδικό (ή τους κωδικούς) και τον τίτλο (ή τους τίτλους) των βιβλίων αυτού του συγγραφέα που υπάρχουν στη βιβλιοθήκη. Το Ν να δηλωθεί ως σταθερά με τιμή 1000. [*Δραστηριότητα ΔΣ4, Κεφάλαιο 4, Τετράδιο Μαθητή*]

(Σκέψη: Εφόσον ο πίνακας δεν είναι ταξινομημένος θα χρησιμοποιηθεί σειριακή αναζήτηση)

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΔΥΑΔΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

- Αρχικά ελέγχουμε αν είναι το στοιχείο αναζήτησης μικρότερο ή μεγαλύτερο από το στοιχείο που βρίσκεται στη μέση του ταξινομημένου πίνακα. Αν είναι μικρότερο, κάνουμε το ίδιο στο πρώτο μισό, αν είναι μεγαλύτερο, τότε ψάχνουμε στο δεύτερο μισό. Κόβουμε δηλαδή κάθε φορά σε κομμάτια τον πίνακά μας.

π.χ.



Σχόλιο: Αν ο πίνακας έχει άρτιο αριθμό στοιχείων, κάνουμε πάλι το ίδιο.

Ερώτηση: Πως θα βρω τη μέση, σαν θέση γνωρίζοντας πόσα στοιχεία έχει ο πίνακας;

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΔΥΑΔΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Γ.42. Δίνονται ως δεδομένα ταξινομημένος πίνακας Table[100] και τιμή αναζήτησης key στον πίνακα. Να γραφεί ο αλγόριθμος δυαδικής αναζήτησης της τιμής στον πίνακα.

Λύση

```
Δεδομένα //Table, Key//
αρχή ← 1
τέλος ← 100
flag ← Ψευδής
pos ← -1
Όσο αρχή <= τέλος και flag = Ψευδής επανάλαβε
        μέση ← (αρχή + τέλος) div 2
        Αν  key > Table[μέση]  τότε
                    αρχή ← μέση + 1
        Αλλιώς_αν  key < Table[μέση]  τότε
                    τέλος ← μέση - 1
        Αλλιώς  ! περίπτωση ισότητας, δηλ. βρέθηκε
                    flag ← Αληθής
                    pos ← μέση
        Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε βρέθηκε, pos
```

! Τώρα η αρχή πάει μετά το μεσαίο στοιχείο, γιατί πριν από αυτό το στοιχείο δεν είχα βρει το key

! Τώρα το τέλος πάει πριν το μεσαίο στοιχείο, γιατί μετά από αυτό το στοιχείο δεν είχα βρει το key

! Αν δεν συμβεί τίποτα από τα προηγούμενα, τότε αναγκαστικά το key βρίσκεται ακριβώς στη θέση μέση

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

- Να δοθεί ο ορισμός της ταξινόμησης Ν στοιχείων $\alpha_1, \alpha_{2, \dots}, \alpha_N$.

Η ταξινόμηση των στοιχείων $\alpha_1, \alpha_{2, \dots}, \alpha_N$ γίνεται με τη μετάθεση της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά $\alpha_{κ1}, \alpha_{κ2, \dots}, \alpha_{κN}$ έτσι ώστε, αν δοθεί μια συνάρτηση διάταξης f, να ισχύει:

$$\mathbf{f(\alpha_{κ1}) \leqslant f(\alpha_{κ2}) \leqslant \dots \leqslant f(\alpha_{κN})} \quad \text{ή} \quad \mathbf{f(\alpha_{κ1}) \geqslant f(\alpha_{κ2}) \geqslant \dots \geqslant f(\alpha_{κN})} .$$

- Αλγόριθμος Φυσαλίδας

Αυτός ο αλγόριθμος βασίζεται στις διαδοχικές αντιμεταθέσεις στοιχείων έτσι ώστε τα στοιχεία με τις μικρότερες τιμές να ανέβουν στις πρώτες θέσεις του πίνακα. Τα μικρότερα στοιχεία ή κατά μία έννοια τα “ελαφρύτερα”, ακολουθούν την πορεία μια φυσαλίδας, δηλαδή πάνε προς τα πάνω (κατά αύξουσα σειρά).

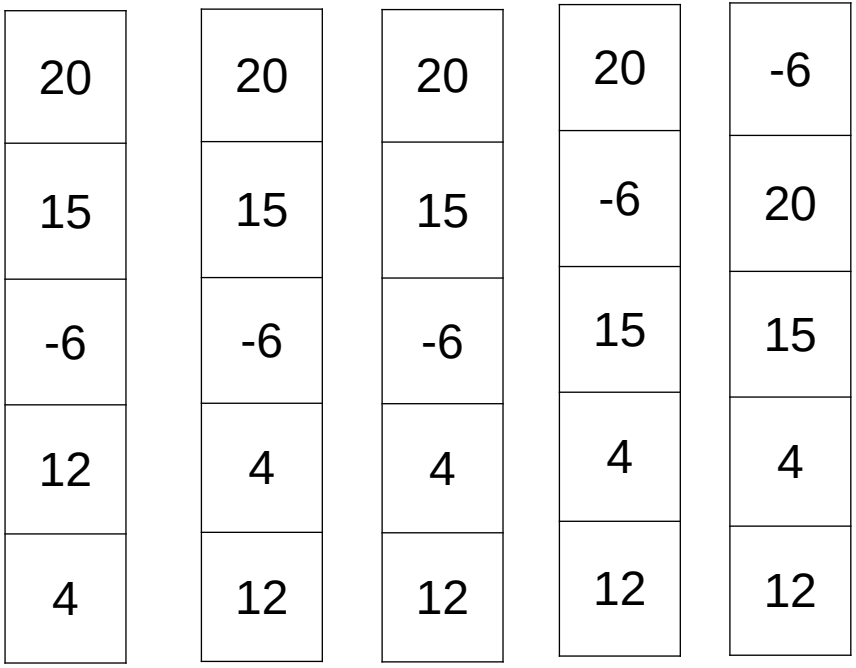
π.χ. Έστω ο μονοδιάστατος πίνακας 5 θέσεων:

20	15	-6	12	4

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

Βήμα 1.

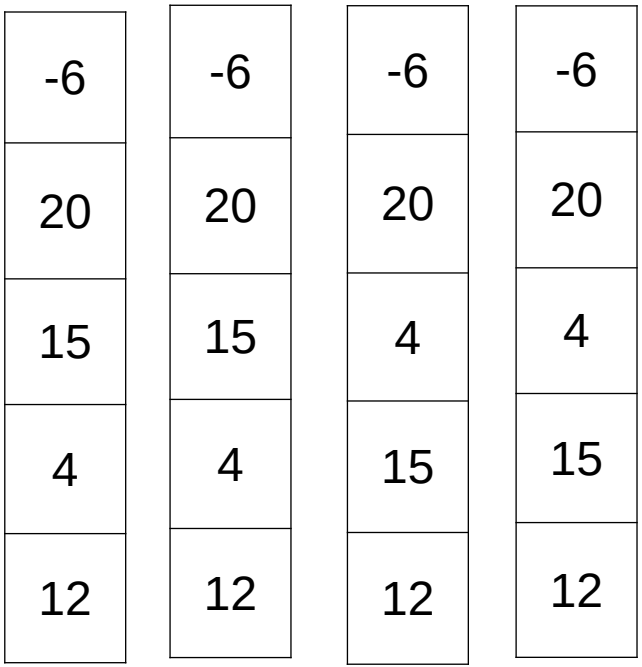
- Ο αλγόριθμος, αρχικά, έχει ως στόχο να βρει το μικρότερο στοιχείο και να το τοποθετήσει στην πρώτη θέση.
- Αρχίζοντας από το τελευταίο στοιχείο του πίνακα, συγκρίνουμε τα στοιχεία ανά δύο και τα αντιμεταθέτουμε, αν χρειαστεί, ώστε το μικρότερο από τα δύο στοιχεία να βρεθεί πρώτο.
- Στον πίνακα του παραδείγματος, συγκρίνουμε το 5^{ο} στοιχείο με το 4^{ο} και τα αντιμεταθέτουμε, ώστε το 4 να ανέβει πριν το 12.
- Έπειτα συγκρίνουμε το 4^{ο} με το 3^{ο} στοιχείο, αλλά δεν χρειάζεται να κάνουμε αντιμετάθεση τιμών, γιατί το -6 βρίσκεται ήδη πριν το 4.
- Στη συνέχεια γίνεται σύγκριση μεταξύ του 3^{ου} και του 2^{ου} στοιχείου και το -6 ανεβαίνει πριν το 15.
- Τέλος, γίνεται σύγκριση του 2^{ου} και του 1^{ου} στοιχείου και το -6 πηγαίνει στην 1η θέση του πίνακα ενώ το 20 στη δεύτερη θέση.



ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

Βήμα 2.

- Στο βήμα 2 δεν ασχολούμαστε με το πρώτο στοιχείο του πίνακα διότι όπως γνωρίζουμε από το προηγούμενο βήμα, αυτό είναι το μικρότερο στοιχείο του πίνακα και βρίσκεται ήδη στην τελική του θέση.
- Ο αλγόριθμος πλέον προσπαθεί να βρει το αμέσως μικρότερο στοιχείο και να το τοποθετήσει στη δεύτερη θέση.
- Αρχίζοντας πάλι από το τελευταίο στοιχείο του πίνακα, συγκρίνουμε τα στοιχεία ανά δύο και τα αντιμεταθέτουμε, όπως νωρίτερα, με τη διαφορά όμως ότι η σύγκριση δεν φθάνει στο πρώτο στοιχείο του πίνακα.
- Η διαδικασία συνεχίζεται για δύο ακόμη φορές. Στο βήμα 3 αρχίζοντας, από το τέλος συγκρίνουμε στοιχεία ανά δύο και τελικά το 12 πάει στην 3η θέση. Στο τελευταίο βήμα 4 αρχίζουμε πάλι από το τέλος τις συγκρίσεις και το 12 φτάνει στην 4η θέση.
- Ο τελικός πίνακας θα είναι:



-6	4	12	15	20

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

Παρατηρήσεις

1. Για να πάει κάθε στοιχείο στην τελική του θέση, έγιναν συνολικά τόσα βήματα όσα τα στοιχεία του πίνακα μείον ένα. Συνεπώς, αν είχαμε Ν στοιχεία θα γινόντουσαν Ν-1 βήματα ή διαφορετικά Ν-1 επαναλήψεις της ίδιας διαδικασίας.
2. Σε κάθε βήμα γινόταν διαφορετικός αριθμός συγκρίσεων των ζευγών των αριθμών. Στο 1ο βήμα έγιναν 4 συγκρίσεις, στο 2ο βήμα 3 συγκρίσεις, στο 3ο έγιναν 2 συγκρίσεις και στο τελευταίο βήμα 1 σύγκριση. Αν προσπαθήσουμε να το εκφράσουμε αλγοριθμικά ως μια επανάληψη, η επανάληψη θα γίνεται τόσες φορές όσες και τα εναπομείναντα αταξινόμητα στοιχεία μείον ένα.

```
Αλγόριθμος Φυσαλίδα
  Δεδομένα //Ν, Π//
  Για i από 2 μέχρι Ν                          ! Η επανάληψη γίνεται Ν-1 φορές
      Για j από Ν μέχρι i με_βήμα -1           ! Η σύγκριση αρχίζει από το τέλος
          Αν Π[j-1] > Π[j]  τότε
              Αντιμετάθεσε Π[j-1], Π[j]
          Τέλος_αν
      Τέλος_επανάληψης
  Τέλος_επανάληψης
```

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

```
Αλγόριθμος Φυσαλίδα
   Δεδομένα //Ν, Π//
   Για i από 2 μέχρι N                          ! Η επανάληψη γίνεται Ν-1 φορές
        Για j από N μέχρι i με_βήμα -1          ! Η σύγκριση αρχίζει από το τέλος
             Αν Π[j-1] > Π[j]  τότε
                  Αντιμετάθεσε Π[j-1], Π[j]
             Τέλος_αν
        Τέλος_επανάληψης
   Τέλος_επανάληψης
Αποτελέσματα //Π//
Τέλος Φυσαλίδα
```

Παρατηρήσεις

1. Η εντολή αντιμετάθεσε, αντιμεταθέτει τις τιμές των στοιχείων. Στη γλώσσα δεν χρησιμοποιούμε αυτή την εντολή, αλλά:

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

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

Γ.44. Να γραφεί πρόγραμμα που θα διαβάζει έναν πίνακα 50 θέσεων με πραγματικούς αριθμούς και θα εμφανίζει τους 5 μεγαλύτερους αριθμούς του πίνακα.

Δύση

ΠΡΟΣΟΧΗ! Στη φθίνουσα ταξινόμηση θέλουμε στην 1η θέση του πίνακα να πάει το μεγαλύτερο στοιχείο. Οπότε μπορούμε να ελέγξουμε αν το τελευταίο στοιχείο Π[j] είναι > από το προτελευταίο Π[j-1] τότε να κάνουμε αντιμετάθεση. Το ίδιο θα ισχύει και για τα άλλα ζεύγη έπειτα.
Εναλλακτικά μπορούμε να ελέγχουμε αν το προτελευταίο Π[j-1] είναι < από το Π[j] τα αντιμεταθέτουμε ώστε α ανέβει προς τα πάνω το μεγαλύτερο.

```
ΠΡΟΓΡΑΜΜΑ Ασκ
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: i, j
   ΠΡΑΓΜΑΤΙΚΕΣ: Π[50], temp
ΑΡΧΗ
   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
         ΔΙΑΒΑΣΕ Π[i]
   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ – ΑΛΓΟΡΙΘΜΟΣ ΤΑΞΙΝΟΜΗΣΗΣ ΦΥΣΑΛΙΔΑΣ

Γ.44. Να γραφεί πρόγραμμα που θα διαβάζει έναν πίνακα 50 θέσεων με πραγματικούς αριθμούς και θα εμφανίζει τους 5 μεγαλύτερους αριθμούς του πίνακα.

Δύση

```
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 50
    ΓΙΑ j ΑΠΟ 50 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1          ! φθίνουσα ταξινόμηση
        ΑΝ Π[j-1] < Π[j]  ΤΟΤΕ                ! ή Π[j] > Π[j-1]
            temp ← Π[j-1]
            Π[j-1] ← Π[j]
            Π[j] ← temp
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΡΑΨΕ Π[i]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```