

ΠΛΗ10

ΕΝΟΤΗΤΑ 2: Αλγόριθμοι και Ψευδογλώσσα

Μάθημα 2.5: Πίνακες

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Πίνακες

1. Μονοδιάστατοι Πίνακες

1. Γενικά
2. Δήλωση Πίνακα
3. Παράδειγμα και Απεικόνιση στη Μνήμη
4. Πρόσβαση και Επεξεργασία Στοιχείων Πίνακα
5. Εντολές Επανάληψης και Μονοδιάστατοι Πίνακες

2. Διδιάστατοι Πίνακες

1. Δήλωση Πίνακα
2. Παράδειγμα και Απεικόνιση στη Μνήμη
3. Επεξεργασία Στοιχείων Πίνακα

3. Πολυδιάστατοι Πίνακες

1. Τριδιάστατοι Πίνακες
2. Μεγαλύτερης Διάστασης Πίνακες

4. Στατικοί Πίνακες

1. Δήλωση Μεγέθους Πίνακα μέσω Σταθεράς

B. Συνήθεις Αλγόριθμοι Πινάκων

1. Διαπέραση Πίνακα

1. Ελάχιστο Στοιχείο Πίνακα
2. Άθροισμα Στοιχείων Πίνακα
3. Αντιγραφή Πινάκων

2. Αναζήτηση Στοιχείου σε Πίνακα

1. Σειριακή Αναζήτηση Στοιχείου
2. Δυαδική Αναζήτηση Στοιχείου

3. Ταξινόμηση Πινάκων

1. Ταξινόμηση με Επιλογή (Selection Sort)
2. Ταξινόμηση με Εισαγωγή (Insertion Sort)
3. Ο αλγόριθμος της φυσαλίδας (Bubble Sort)
4. ...και άλλοι αλγόριθμοι ταξινόμησης!

Γ. Ασκήσεις

1. Μέσος Όρος Στοιχείων Πίνακα
2. Μέγιστος και 2^{ος} μέγιστος αριθμός
3. Μέγιστος Διδιάστατου Πίνακα
4. Ανάστροφος Πίνακας
5. Άθροισμα Διδιάστατων Πινάκων
6. Γινόμενο Διδιάστατων Πινάκων



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

1. Γενικά

- Ο πίνακας είναι ένα εργαλείο με το οποίο αποθηκεύουμε πολλές ίδιου τύπου μεταβλητές στην μνήμη του υπολογιστή και έπειτα τις μεταχειριζόμαστε με έναν κοινό τρόπο
- Είναι κοινό εργαλείο σε όλες τις γλώσσες προγραμματισμού και η χρήση τους απαιτείται όταν θέλουμε να κατασκευάσουμε περίπλοκους αλγόριθμους.
 - Αποτελεί στην πραγματικότητα την πρώτη μας δομή δεδομένων!
- Οι πίνακες χωρίζονται σε κατηγορίες ανάλογα με τη διάστασή τους:
 - Μονοδιάστατος πίνακας: Είναι πίνακας που αποθηκεύει N μεταβλητές κοινού τύπου δεδομένων
 - Διδιάστατος πίνακας: Είναι πίνακας που αποθηκεύει $M \times N$ μεταβλητές κοινού τύπου δεδομένων
 - Πολυδιάστατοι πίνακες: Αντίστοιχα μπορούμε να ορίσουμε τριδιάστατους, τετραδιάστατους κ.ο.κ. Πίνακες δεδομένων.



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

2. Δήλωση Πίνακα

- Ένας μονοδιάστατος πίνακας δηλώνεται στο τμήμα δήλωσης δεδομένων ως:

ΔΕΔΟΜΕΝΑ :

ΟΝΟΜΑ : **ARRAY**[1..N] **OF** ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ ;

- Όπου δηλώνεται ένας μονοδιάστατος πίνακας με N στοιχεία του ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ με την ονομασία ΟΝΟΜΑ.
- Με την δήλωση του πίνακα έχουμε (διαδοχικά αποθηκευμένες στην μνήμη) N μεταβλητές ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ με ονόματα:
 - ΟΝΟΜΑ[1]
 - ΟΝΟΜΑ[2]
 - ΟΝΟΜΑ[3]
 -
 - ΟΝΟΜΑ[N]
- τις οποίες μπορούμε να διαχειριστούμε αντίστοιχα όπως χειριζόμαστε μια μεταβλητή.



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

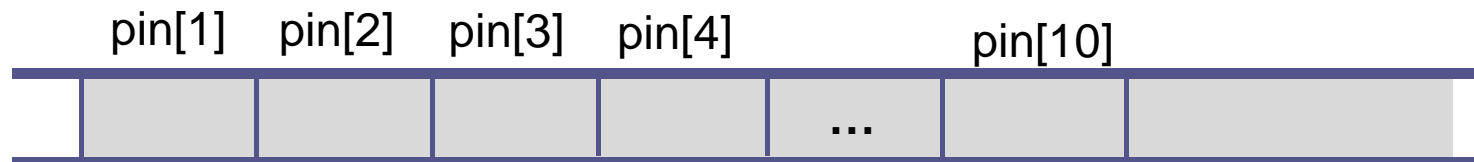
3. Παράδειγμα και απεικόνιση στη Μνήμη

- Έστω η εντολή δήλωσης:

ΔΕΔΟΜΕΝΑ:

```
pin: ARRAY[1..10] OF INTEGER;
```

- Τότε στην μνήμη σε 10 διαδοχικές θέσεις έχουμε ότι δεσμεύεται χώρος για 10 ακέραιες μεταβλητές:
- Η εικόνα της μνήμης είναι:





A. Θεωρία

1. Μονοδιάστατοι Πίνακες

4. Πρόσβαση και επεξεργασία στοιχείων του Πίνακα

- Ένα πίνακας 5 ακεραίων με όνομα P δηλώνεται ως εξής:

ΔΕΔΟΜΕΝΑ:

```
P: ARRAY[1..5] OF INTEGER;
```

- Με την δήλωση έχουμε τις μεταβλητές P[1],P[2],P[3],P[4],P[5] σε διαδοχικές θέσεις στην μνήμη.
- Τις οποίες μπορούμε να διαχειριστούμε ως ακέραιες μεταβλητές, όπως για π.χ. στο ακόλουθο τμήμα κώδικα:

ΑΡΧΗ

```
P[1] := 5;  
P[2] := 10;  
P[3] := 15;  
P[4] := 20;  
P[5] := 25;  
ΤΥΠΩΣΕ ( P[1] + P[2] + P[3] + P[4] + P[5] );
```

ΤΕΛΟΣ

- που θα εκτυπώσει το αποτέλεσμα 75.

A. Θεωρία

1. Μονοδιάστατοι Πίνακες

5. Εντολές Επανάληψης και Μονοδιάστατοι Πίνακες

- Στο ακόλουθο παράδειγμα διαβάζουμε 10 ακεραίους και τους αποθηκεύουμε σε έναν πίνακα. Έπειτα υπολογίζουμε το διπλάσιο κάθε στοιχείου και εκτυπώνουμε τον πίνακα

ΑΛΓΟΡΙΘΜΟΣ monodiastatos

ΣΤΑΘΕΡΕΣ

N=10;

ΔΕΔΟΜΕΝΑ

PIN: ARRAY[1..N] OF INTEGER;

i: INTEGER;

ΑΡΧΗ

ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ

ΔΙΑΒΑΣΕ(PIN[i]);

ΓΙΑ-ΤΕΛΟΣ

ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ

PIN[i]:=2*PIN[i];

ΓΙΑ-ΤΕΛΟΣ

ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ

ΤΥΠΩΣΕ (PIN[i]);

ΓΙΑ-ΤΕΛΟΣ

ΤΕΛΟΣ

Άσκηση: Τροποποιήστε το διάβασμα δεδομένων και την εκτύπωση του προγράμματος έτσι ώστε να είναι πιο «κομψή».



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

5. Εντολές Επανάληψης και Μονοδιάστατοι Πίνακες

- Είναι εξαιρετικά συνηθισμένο να έχουμε κάποια δεδομένα σε έναν πίνακα, στα οποία να κάνουμε μια επεξεργασία και να λύνουμε ένα πρόβλημα.
- Το σημαντικό είναι ότι κάνουμε μία (ή παραπάνω) διαπεράσεις των στοιχείων του πίνακα προκειμένου να εξάγουμε το ζητούμενο συνήθως με την εντολή επανάληψης ΓΙΑ...ΓΙΑ-ΤΕΛΟΣ.
- Έτσι οι ενέργειες που κάναμε στο προηγούμενο παράδειγμα είναι εξαιρετικά συνηθισμένες σε ένα πρόγραμμα σε μια διαδικαστική γλώσσα προγραμματισμού:
 - **ΔΙΑΒΑΣΕ τα στοιχεία του πίνακα**
 - **Επεξεργάσου τα στοιχεία του πίνακα**
 - **ΤΥΠΩΣΕ το αποτέλεσμα**
- Έτσι θα δώσουμε βάρος σε **αλγόριθμους** που κάνουν συνηθισμένες υπολογιστικές διαδικασίες:
 - Τον εντοπισμό ενός στοιχείου με μια ιδιότητα (π.χ. **ελάχιστος, μέγιστος**)
 - Τον υπολογισμό μιας ποσότητας που σχετίζει τα στοιχεία του πίνακα (π.χ. **άθροισμα, γινόμενο, μέσος όρος**)
- Και θα δώσουμε ιδιαίτερο βάρος σε αυτό και το επόμενο μάθημα σε δύο περίφημα προβλήματα:
 - Την **αναζήτηση ενός στοιχείου** σε έναν πίνακα
 - Την **ταξινόμηση των στοιχείων** ενός πίνακα.



A. Θεωρία

2. Διδιάστατοι Πίνακες

1. Δήλωση Πίνακα

- Ένας διδιάστατος πίνακας δηλώνεται στο τμήμα δήλωσης δεδομένων ως:

ΔΕΔΟΜΕΝΑ :

ΟΝΟΜΑ : ARRAY[1..M,1..N] OF ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ ;

- Όπου δηλώνεται ένας διδιάστατος πίνακας με $M \times N$ δεδομένα του ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ με την ονομασία ΟΝΟΜΑ.
- Με την δήλωση του πίνακα έχουμε (διαδοχικά αποθηκευμένες στην μνήμη) $M \times N$ μεταβλητές ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ με ονόματα:
 - ΟΝΟΜΑ[1,1],ΟΝΟΜΑ[1,2],...,ΟΝΟΜΑ[1,N]
 - ΟΝΟΜΑ[2,1],ΟΝΟΜΑ[2,2],...,ΟΝΟΜΑ[2,N]
 - ΟΝΟΜΑ[3,1],ΟΝΟΜΑ[3,2],...,ΟΝΟΜΑ[3,N]
 -
 - ΟΝΟΜΑ[M,1],ΟΝΟΜΑ[M,2],...,ΟΝΟΜΑ[M,N]
- τις οποίες μπορούμε να διαχειριστούμε αντίστοιχα όπως χειριζόμαστε μια μεταβλητή.



A. Θεωρία

2. Διδιάστατοι Πίνακες

2. Παράδειγμα και απεικόνιση στη Μνήμη

- Έστω η εντολή δήλωσης:

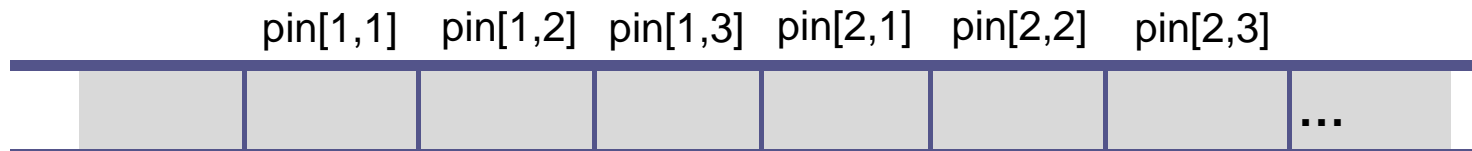
ΔΕΔΟΜΕΝΑ:

```
pin: ARRAY[1..2,1..3] OF INTEGER;
```

- Με την οποία μπορούμε να διαχειριστούμε τις ακέραιες μεταβλητές:
 - pin[1,1], pin[1,2], pin[1,3]
 - pin[2,1], pin[2,2], pin[2,3]
- Οπτικά θα έχουμε την εξής εικόνα του πίνακα για να γράψουμε τους αλγόριθμούς μας:

pin[1,1]	pin[1,2]	pin[1,3]
pin[2,1]	pin[2,2]	pin[2,3]

- Αλλά στην πραγματικότητα, στην μνήμη, σε 6 διαδοχικές θέσεις, έχουμε ότι δεσμεύεται χώρος για 6 ακέραιες μεταβλητές:
- Η εικόνα της μνήμης είναι:





A. Θεωρία

2. Διδιάστατοι Πίνακες

3. Επεξεργασία Στοιχείων Διδιάστατου Πίνακα

- Όταν θέλουμε να επεξεργαστούμε έναν διδιάστατο πίνακα συνήθως κάνουμε μία επανάληψη πάνω στις γραμμές και μία εμφωλιασμένη επανάληψη πάνω στις στήλες.
- Για παράδειγμα έστω ότι θέλουμε να κατασκευάσουμε έναν 3x4 πίνακα που να αποθηκεύει στην θέση $[i,j]$ το αποτέλεσμα της πρόσθεσης $i+j$ όπως φαίνεται στο ακόλουθο σχήμα:
 - Αυτό μπορεί να γίνει με τον εξής κώδικα:

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

ΔΕΔΟΜΕΝΑ

```
P: ARRAY[1..3,1..4] OF INTEGER;  
i, j: INTEGER;
```

ΑΡΧΗ

```
ΓΙΑ i:=1 ΕΩΣ 3 ΕΠΑΝΑΛΑΒΕ  
    ΓΙΑ j:=1 ΕΩΣ 4 ΕΠΑΝΑΛΑΒΕ  
        P[i,j]:=i+j;  
    ΓΙΑ-ΤΕΛΟΣ  
ΓΙΑ-ΤΕΛΟΣ  
ΤΕΛΟΣ
```

Άσκηση: Δημιουργήστε ένα ολοκληρωμένο πρόγραμμα που να εκτυπώνει «κομψά» τα περιεχόμενα του παραπάνω πίνακα.



A. Θεωρία

3. Πολυδιάστατοι Πίνακες

1. Τριδιάστατοι Πίνακες

- Ένας τριδιάστατος πίνακας δηλώνεται στο τμήμα δήλωσης δεδομένων ως:

ΔΕΔΟΜΕΝΑ:

ΟΝΟΜΑ: ARRAY[1..M,1..N,1..R,1..T] OF ΤΥΠΟΣ-ΔΕΔΟΜΕΝΩΝ

- Όπου δηλώνεται ένας τριδιάστατος πίνακας με $M \times N \times K$ δεδομένα του ΤΥΠΟΥ-ΔΕΔΟΜΕΝΩΝ με την ονομασία ΟΝΟΜΑ.
- Αντίστοιχα οι μεταβλητές θα χρησιμοποιούνται ως ΟΝΟΜΑ[I,J,K] με:
 - Το $1 \leq I \leq M$
 - Το $1 \leq J \leq N$
 - Το $1 \leq K \leq R$
- Και όταν θα διαχειριστούμε τα δεδομένα αυτό θα γίνει με τριπλή επανάληψη

```
ΓΙΑ i:=1 ΕΩΣ M ΕΠΑΝΑΛΑΒΕ  
  ΓΙΑ j:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ  
    ΓΙΑ k:=1 ΕΩΣ R ΕΠΑΝΑΛΑΒΕ  
      ...
```



A. Θεωρία

3. Πολυδιάστατοι Πίνακες

2. Μεγαλύτερης Διάστασης Πίνακες

- Αντίστοιχα μπορούν να οριστούν πολυδιάστατοι πίνακες.
- Π.χ. Ένας τετραδιάστατος πίνακας θα δηλώνεται:

ΔΕΔΟΜΕΝΑ:

ΟΝΟΜΑ: **ARRAY**[1..M,1..N,1..R,1..T] **OF** ΤΥΠΟΣ-ΔΕΔΟΜΕΝΩΝ

- Και θα απαιτεί τετραπλό ΓΙΑ...ΕΠΑΝΕΛΑΒΕ
- Ενώ ένας πενταδιάστατος πίνακας θα δηλώνεται:

ΔΕΔΟΜΕΝΑ:

ΟΝΟΜΑ: **ARRAY**[1..M,1..N,1..R,1..T,1..S] **OF** ΤΥΠΟΣ-ΔΕΔΟΜΕΝΩΝ

- Και θα απαιτεί πενταπλό ΓΙΑ...ΕΠΑΝΕΛΑΒΕ
- Ωστόσο αυτά δεν πρόκειται να μας ζητηθούν σε ασκήσεις.



A. Θεωρία

4. Στατικοί Πίνακες

1. Δήλωση Διάστασης Πίνακα μέσω Σταθεράς

- Έχουμε τρεις τρόπους για να δηλώσουμε τη διάσταση (μέγεθος) του πίνακα:.

Α' τρόπος:

- «Καρφωτά» μέσω ενός αριθμού.

Π.χ. στο τμήμα δήλωσης δεδομένων γράφουμε:

ΔΕΔΟΜΕΝΑ

```
PIN: ARRAY[1..10] OF INTEGER;
```

Β' τρόπος:

- Έμμεσα μέσω μίας σταθεράς.

Π.χ. στο τμήμα δήλωσης δεδομένων γράφουμε:

ΣΤΑΘΕΡΕΣ

```
N=10;
```

ΔΕΔΟΜΕΝΑ

```
PIN: ARRAY[1..N] OF INTEGER;
```




A. Θεωρία

4. Στατικοί Πίνακες

1. Δήλωση Διάστασης Πίνακα μέσω Σταθεράς

- Έχουμε τρεις τρόπους για να δηλώσουμε τη διάσταση (μέγεθος) του πίνακα:

Γ' τρόπος:

- Δηλώνουμε το μέγιστο μέγεθος πίνακα μέσω σταθεράς και έπειτα έχουμε δεύτερη μεταβλητή για το πλήθος των θέσεων που χρησιμοποιούμε.

Παράδειγμα!

ΣΤΑΘΕΡΕΣ

```
MAX_N=10;
```

ΔΕΔΟΜΕΝΑ

```
PIN: ARRAY[1..MAX_N] OF INTEGER;
```

```
N: INTEGER;
```

- Σημαντικό: Τον πρώτο τρόπο τον αποφεύγουμε, τον δεύτερο τρόπο τον χρησιμοποιούμε όταν γνωρίζουμε το μέγεθος του πίνακα είναι συγκεκριμένο στον αλγόριθμο και τον τρίτο τρόπο όταν θέλουμε το μέγεθος του πίνακα να ορίζεται από τον χρήστη.

Άσκηση: Κατασκευάστε πρόγραμμα με τον τρίτο τρόπο που ορίζει την μέγιστη διάσταση σε 10, έπειτα διαβάζει από την είσοδο με αμυντικό προγραμματισμό το N (από το 5 έως το 10) και έπειτα διαβάζει N ακεραίους και τους τυπώνει στην οθόνη.



Β. Συνήθεις Αλγόριθμοι Πινάκων

1. Διαπέραση Πίνακα

1. Ελάχιστο Στοιχείο Πίνακα

- Ο ακόλουθος αλγόριθμος υπολογίζει το ελάχιστο στοιχείο ενός πίνακα ακεραίων PIN με N στοιχεία και το αποθηκεύει σε μία μεταβλητή min:

```
min:=PIN[1];
```

```
ΓΙΑ i:=2 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
```

```
    ΕΑΝ (PIN[i] < min) ΤΟΤΕ
```

```
        min:=PIN[i];
```

```
    ΕΑΝ-ΤΕΛΟΣ
```

```
ΓΙΑ-ΤΕΛΟΣ
```

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν $PIN=[6,4,2,7,3]$.

Άσκηση 2: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)

Άσκηση 3: Τροποποιήστε το πρόγραμμα έτσι ώστε να βρίσκει το μέγιστο στοιχείο του πίνακα.



Β. Συνήθεις Αλγόριθμοι Πινάκων

1. Διαπέραση Πίνακα

2. Άθροισμα Στοιχείων Πίνακα

- Το ακόλουθο τμήμα κώδικα υπολογίζει το άθροισμα των στοιχείων ενός πίνακα ακεραίων PIN με N στοιχεία και το αποθηκεύει σε μία μεταβλητή sum:

```
sum:=0;
```

```
ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
```

```
    sum:=sum+pin[i];
```

```
ΓΙΑ-ΤΕΛΟΣ
```

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν $PIN=[6,4,2,7,3]$.

Άσκηση 2: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)

Άσκηση 3: Τροποποιήστε το πρόγραμμα έτσι ώστε να βρίσκει το γινόμενο των στοιχείων του πίνακα.



B. Συνήθεις Αλγόριθμοι Πινάκων

1. Διαπέραση Πίνακα

3. Αντιγραφή Πίνακα

- Το ακόλουθο τμήμα κώδικα αντιγράφει τον πίνακα ακεραίων A στον πίνακα ακεραίων B:

```
ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ  
    B[i]:=A[i];  
ΓΙΑ-ΤΕΛΟΣ
```

Άσκηση 1: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων του A, αντιγραφή στον B)

Άσκηση 2: Δημιουργήστε ένα δεύτερο πρόγραμμα το οποίο:

- Δημιουργεί τρεις πίνακες A,B και C, ο καθένας N θέσεων.
- Διαβάζει τα στοιχεία του A
- Αντιγράφει στον πίνακα A στον B
- Διπλασιάζει κάθε στοιχείο του B
- Αντιγράφει στον πίνακα B στον C
- Μειώνει στο μισό κάθε στοιχείο του C.
- Τυπώνει τους πίνακες A,B,C.



Β. Συνήθεις Αλγόριθμοι Πινάκων

2. Αναζήτηση Στοιχείου

1. Σειριακή Αναζήτηση

- Διατύπωση Προβλήματος: Δίνεται πίνακας ακεραίων PIN και στοιχείο x. Υπάρχει το στοιχείο x στον πίνακα PIN;

```
check:=FALSE; /* BOOLEAN */
```

```
ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
```

```
    ΕΑΝ (x=PIN[i]) ΤΟΤΕ
```

```
        check:=TRUE;
```

```
    ΕΑΝ-ΤΕΛΟΣ
```

```
ΓΙΑ-ΤΕΛΟΣ
```

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν $PIN=[6,4,2,7,3]$, $x=7$.

Άσκηση 2: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)

Άσκηση 3: Τροποποιήστε το πρόγραμμα έτσι ώστε να βρίσκει και την θέση του στοιχείου στον πίνακα.



Β. Συνήθεις Αλγόριθμοι Πινάκων

2. Αναζήτηση Στοιχείου

2. Δυαδική Αναζήτηση

- Διατύπωση Προβλήματος: Δίνεται **ταξινομημένος** πίνακας ακεραίων PIN (σε αύξουσα σειρά) και στοιχείο x. Υπάρχει το στοιχείο x στον πίνακα PIN;

```
check:=FALSE;  
start:=1;  
finish:=N;  
ΕΝΟΣΩ (start<=finish AND check=FALSE)  
ΕΠΑΝΑΛΑΒΕ  
    middle:=(start+finish) DIV 2;  
    ΕΑΝ (x=PIN[middle]) ΤΟΤΕ  
        check:=TRUE;  
    ΑΛΛΙΩΣ  
        ΕΑΝ (x>PIN[middle]) ΤΟΤΕ  
            start:=middle+1;  
        ΑΛΛΙΩΣ  
            finish:=middle-1;  
    ΕΑΝ-ΤΕΛΟΣ  
ΕΑΝ-ΤΕΛΟΣ  
ΕΝΟΣΩ-ΤΕΛΟΣ
```

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν PIN=[2,4,6,8,10,12,14,16,18,20] , x=8.

Άσκηση 2: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν PIN=[2,4,6,8,10,12,14,16,18,20] , x=15.

Άσκηση 3: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)



B. Συνήθεις Αλγόριθμοι Πινάκων

3. Ταξινόμηση Πίνακα

1. Ταξινόμηση με Επιλογή (Selection Sort)

- Διατύπωση Προβλήματος: Δίνεται πίνακας ακεραίων PIN. Ζητείται να ταξινομηθούν τα στοιχεία του σε αύξουσα σειρά με τον αλγόριθμο ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (SELECTION SORT).

```
ΓΙΑ i:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
    /* Αναζήτηση ελαχίστου στις
       θέσεις j=i+1...N */
    pos:=i;
    ΓΙΑ j:=i+1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
        ΕΑΝ (PIN[j]<PIN[pos]) ΤΟΤΕ
            pos:=j;
        ΕΑΝ-ΤΕΛΟΣ
    ΓΙΑ-ΤΕΛΟΣ

    /* Ανταλλαγή του ελαχίστου με
       το PIN[i] */
    temp:=PIN[i];
    PIN[i]:=PIN[pos];
    PIN[pos]:=temp;
ΓΙΑ-ΤΕΛΟΣ
```

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν PIN=[5,8,4,2,9,3,1,4,6]

Άσκηση 2: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)



Β. Συνήθεις Αλγόριθμοι Πινάκων

3. Ταξινόμηση Πίνακα

2. Ταξινόμηση με Εισαγωγή (Insertion Sort)

- Διατύπωση Προβλήματος: Δίνεται πίνακας ακεραίων PIN. Ζητείται να ταξινομηθούν τα στοιχεία του σε αύξουσα σειρά με τον αλγόριθμο ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΙΣΑΓΩΓΗ (INSERTION SORT).

ΓΙΑ $i:=2$ **ΕΩΣ** N **ΕΠΑΝΑΛΑΒΕ**

$j:=i;$

$check:=FALSE;$

ΕΝΟΣΩ $j \geq 2$ **AND** $check=FALSE$ **ΕΠΑΝΑΛΑΒΕ**

ΕΑΝ $(PIN[j] < PIN[j-1])$ **ΤΟΤΕ**

$temp:=PIN[j];$

$PIN[j]:=PIN[j-1];$

$PIN[j-1]:=temp;$

ΑΛΛΙΩΣ

$check:=TRUE;$

ΕΑΝ-ΤΕΛΟΣ

$j:=j-1;$

ΕΝΟΣΩ-ΤΕΛΟΣ

ΓΙΑ-ΤΕΛΟΣ

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν $PIN=[5,8,4,2,9,3,1,7,6]$

Άσκηση 2: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)



Β. Συνήθεις Αλγόριθμοι Πινάκων

3. Ταξινόμηση Πίνακα

3. Ταξινόμηση Φυσαλίδας (Bubble Sort)

- Διατύπωση Προβλήματος: Δίνεται πίνακας ακεραίων PIN. Ζητείται να ταξινομηθούν τα στοιχεία του σε αύξουσα σειρά με τον αλγόριθμο ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (BUBBLE SORT).

ΓΙΑ $i := 1$ **ΕΩΣ** N **ΕΠΑΝΑΛΑΒΕ**

ΓΙΑ $j := N$ **ΕΩΣ** $i + 1$ **ΜΕ-ΒΗΜΑ** -1 **ΕΠΑΝΑΛΑΒΕ**

ΕΑΝ ($PIN[j] < PIN[j-1]$) **ΤΟΤΕ**

$temp := PIN[j];$

$PIN[j] := PIN[j-1];$

$PIN[j-1] := temp;$

ΕΑΝ-ΤΕΛΟΣ

ΓΙΑ-ΤΕΛΟΣ

ΓΙΑ-ΤΕΛΟΣ

Άσκηση 1: «Τρέξτε» ένα παράδειγμα εκτέλεσης με το χέρι, αν $PIN = [5, 8, 4, 2, 9, 3, 1, 7, 6]$

Άσκηση 2: Δημιουργήστε στον μεταγλωττιστή, ένα ολοκληρωμένο πρόγραμμα που να αναδεικνύει την λειτουργία του αλγορίθμου (εισαγωγή δεδομένων, κατάλληλη εκτύπωση)



B. Συνήθεις Αλγόριθμοι Πινάκων

3. Ταξινόμηση Πίνακα

4. ... και άλλοι αλγόριθμοι ταξινόμησης

- Και εδώ σταματάμε για την ώρα!
 - Το πεδίο των αλγορίθμων είναι ανεξάντλητο!

- Απομένουν δύο σημαντικοί ακόμη αλγόριθμοι ταξινόμησης:
 - Ο αλγόριθμος ταξινόμησης με συγχώνευση (MergeSort)
 - Ο αλγόριθμος γρήγορης ταξινόμησης (QuickSort)

- που θα μελετήσουμε στο επόμενο μάθημα, έχοντας στα χέρια μας και το προγραμματιστικό εργαλείο των συναρτήσεων.



Γ. Ασκήσεις

Εφαρμογή 1: Μέσος Όρος Στοιχείων Πίνακα

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει έναν αριθμό N μεταξύ του 5 και του 10 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει N ακέραιους αριθμούς και να τους αποθηκεύει σε έναν πίνακα N θέσεων.
- Έπειτα να υπολογίζει τον μέσο όρο των N αριθμών και να τυπώνει το αποτέλεσμα στην οθόνη.



Γ. Ασκήσεις

Εφαρμογή 2: Μέγιστος και 2^{ος} μέγιστος αριθμός

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει έναν αριθμό N μεταξύ του 5 και του 10 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει N ακέραιους αριθμούς και να τους αποθηκεύει σε έναν πίνακα N θέσεων.
- Έπειτα να υπολογίζει το μεγαλύτερο και το δεύτερο μεγαλύτερο αριθμό από αυτούς που είναι αποθηκευμένοι στον πίνακα και να τους τυπώνει στην οθόνη.



Γ. Ασκήσεις

Εφαρμογή 3: Μέγιστος Διδιάστατου Πίνακα

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει τον αριθμό M μεταξύ του 2 και του 5 και τον αριθμό N μεταξύ του 3 και του 6 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει $M \times N$ ακέραιους αριθμούς και να τους αποθηκεύει σε έναν πίνακα $M \times N$ θέσεων.
- Έπειτα να υπολογίζει τον ελάχιστο του πίνακα και να τον τυπώνει στην οθόνη.



Γ. Ασκήσεις

Εφαρμογή 4: Ανάστροφος Τετραγωνικού Πίνακα

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει τον αριθμό N μεταξύ του 2 και του 5 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από τον χρήστη να αρχικοποιήσει έναν $N \times N$ πίνακα με τα στοιχεία του πίνακα.
- Έπειτα να υπολογίζει τον ανάστροφο του πίνακα και να τον τυπώνει στην οθόνη.

Σημείωση:

- Η άσκηση αυτή απαιτεί την γνώση του ανάστροφου πίνακα από τα μαθηματικά. Σε περίπτωση έλλειψης υποβάθρου συμβουλευθείτε την ηχογράφηση.



Γ. Ασκήσεις

Εφαρμογή 5: Πρόσθεση Τετραγωνικών Πινάκων

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει τον αριθμό N μεταξύ του 2 και του 5 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από τον χρήστη να αρχικοποιήσει δύο $N \times N$ πίνακες (ονομάστε τους A και B).
- Το πρόγραμμα να υπολογίζει το άθροισμα των πινάκων (ονομάστε το C) και να το τυπώνει στην οθόνη.

Σημείωση:

- Η άσκηση αυτή απαιτεί την γνώση του αθροίσματος πινάκων από τα μαθηματικά. Σε περίπτωση έλλειψης υποβάθρου συμβουλευθείτε την ηχογράφηση.



Γ. Ασκήσεις

Εφαρμογή 6: Γινόμενο Τετραγωνικών Πινάκων

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει τον αριθμό N μεταξύ του 2 και του 5 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από τον χρήστη να αρχικοποιήσει δύο $N \times N$ πίνακες (ονομάστε τους A και B).
- Το πρόγραμμα να υπολογίζει το γινόμενο των πινάκων (ονομάστε το C) και να το τυπώνει στην οθόνη.

Σημείωση:

- Η άσκηση αυτή απαιτεί την γνώση του γινόμενου πινάκων από τα μαθηματικά. Σε περίπτωση έλλειψης υποβάθρου συμβουλευθείτε την ηχογράφηση.