



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
Πανεπιστημιούπολη Σερρών

Προγραμματισμός Ι (Θ)

Δρ. Δημήτρης Βαρσάμης
Αναπληρωτής Καθηγητής

Οκτώβριος 2019

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι (Θ)

- 1 Δομές δεδομένων - Πίνακες
- 2 Αριθμητικοί Πίνακες
 - Μονοδιάστατοι Πίνακες
- 3 Ασκήσεις
- 4 Λύσεις Ασκήσεων

Contents

- 1 Δομές δεδομένων - Πίνακες
- 2 Αριθμητικοί Πίνακες
 - Μονοδιάστατοι Πίνακες
- 3 Ασκήσεις
- 4 Λύσεις Ασκήσεων

Δομές δεδομένων - Πίνακες

- Οι πίνακες είναι μια στατική δομή δεδομένων
- Ο πίνακας δίνει τη δυνατότητα στον προγραμματιστή να δημιουργήσει μια δομή δεδομένων στην οποία μπορεί να αποθηκεύσει (καταχωρίσει) πολλές τιμές.
- Ο προγραμματιστής αναφέρεται και διαχειρίζεται τα δεδομένα τα οποία έχουν αποθηκευτεί σε πίνακα με τη βοήθεια ενός ονόματος και των δεικτών του πίνακα.
- Οι δείκτες του πίνακα σε C ξεκινούν από το μηδέν και είναι μια αύξουσα ακολουθία. Η τιμή του δείκτη μας δείχνει την θέση του πίνακα.

Δομές δεδομένων - Πίνακες

- Οι πίνακες μπορούν να έχουν πολλές διαστάσεις
 - ▶ Οι μονοδιάστατοι πίνακες οι οποίοι έχουν ένα δείκτη
 - ▶ Οι δισδιάστατοι πίνακες οι οποίοι έχουν δυο δείκτες
 - ▶ Οι πολυδιάστατοι πίνακες οι οποίοι έχουν αριθμό δεικτών αντίστοιχο με τις διαστάσεις του πίνακα.
- Η διαχείριση και επεξεργασία των στοιχείων ενός πίνακα γίνεται με την βοήθεια επαναληπτικών εντολών
- Η πιο κατάλληλη επαναληπτική εντολή για την επεξεργασία πινάκων είναι η εντολή **for**. (είναι προφανές ότι για την επεξεργασία πινάκων μπορούμε να χρησιμοποιήσουμε όποια εντολή επανάληψης θέλουμε)

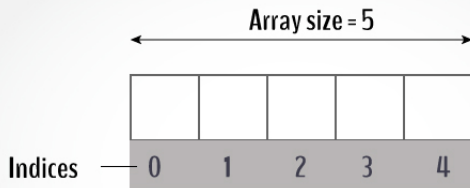
Δομές δεδομένων - Πίνακες

- Τα στοιχεία ενός πίνακα πρέπει να είναι ιδίου τύπου
- Ένας πίνακας μπορεί να δηλωθεί σύμφωνα με τους γνωστούς τύπους δεδομένων της C, δηλαδή, **int**, **float**, **char** κ.α.
- Οι πιο συνηθισμένοι τύποι πινάκων που χρησιμοποιούνται είναι οι αριθμητικοί (arrays) και οι αλφαριθμητικοί ή συμβολοσειρές (strings)

Contents

- 1 Δομές δεδομένων - Πίνακες
- 2 Αριθμητικοί Πίνακες
 - Μονοδιάστατοι Πίνακες
- 3 Ασκήσεις
- 4 Λύσεις Ασκήσεων

Μονοδιάστατοι Πίνακες



C Arrays

Μονοδιάστατοι Πίνακες - Δήλωση

Δήλωση μονοδιάστατου πίνακα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 |     int A[100];
5 |     return 0;
6 | }
```

- Η δήλωση του πίνακα γίνεται όπως των απλών μεταβλητών με την προσθήκη του μεγέθους του πίνακα το οποίο περικλείεται σε αγκύλες

Μονοδιάστατοι Πίνακες - Δήλωση

Δήλωση μονοδιάστατου πίνακα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #define N 10
4 | int main() {
5 |     int A[N];
6 |     return 0;
7 | }
```

- Ένας ορθός τρόπος να δηλώνουμε το μέγεθος του πίνακα είναι με την οδηγία `#define` με την οποία ορίζουμε μια σταθερά

Μονοδιάστατοι Πίνακες - Εμφάνιση I

Εμφάνιση στοιχείων μονοδιάστατου πίνακα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #define N 5
4 | int main() {
5 |     int i, A[N];
6 |     for (i=0; i<N; i=i+1) {
7 |         printf("\t%d", A[i]);
8 |     }
9 |     printf("\n");
10 |    return 0;
11 | }
```

- Για την εμφάνιση των στοιχείων του πίνακα χρησιμοποιούμε την εντολή επανάληψης **for**.

Μονοδιάστατοι Πίνακες - Εμφάνιση II

- Ο δείκτης i ξεκινά από την τιμή μηδέν και φτάνει μέχρι την τιμή $N - 1$.

0	0	48	0	48
---	---	----	---	----

- Αν δεν δώσουμε τιμές στον πίνακα τότε παίρνει τιμές από τα "σκουπίδια" της μνήμης

Μονοδιάστατοι Πίνακες - Αρχικοποίηση - Εμφάνιση I

Αρχικοποίηση και εμφάνιση στοιχείων μονοδιάστατου πίνακα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #define N 5
4 | int main() {
5 |     int i, A[N]={10,100,5,15,25};
6 |     for(i=0;i<N;i=i+1) {
7 |         printf("\t%d",A[i]);
8 |     }
9 |     printf("\n");
10 |    return 0;
11 | }
```

Μονοδιάστατοι Πίνακες - Αρχικοποίηση - Εμφάνιση II

10	100	5	15	25
----	-----	---	----	----

Μονοδιάστατοι Πίνακες - Εισαγωγή - Εμφάνιση I

Εισαγωγή και εμφάνιση στοιχείων μονοδιάστατου πίνακα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #include <math.h>
4 | #define N 5
5 | int main() {
6 |     int A[N], B[N], i;
7 |     for (i=0; i<N; i=i+1) {
8 |         A[i]=pow(i, 2);
9 |         B[i]=2*A[i];
10 |    }
11 |    for (i=0; i<N; i=i+1) {
12 |        printf("\t%d", A[i]);
13 |    }
14 |    printf("\n\n");
```

Μονοδιάστατοι Πίνακες - Εισαγωγή - Εμφάνιση

II

```
15 |     for (i=0;i<N;i=i+1) {  
16 |         printf("\t%d\n",B[i]);  
17 |     }  
18 |     return 0;  
19 | }
```

0	1	4	9	16
---	---	---	---	----

0

2

8

18

32

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία I

Εκχώρηση τιμών και επεξεργασία στοιχείων μονοδιάστατου πίνακα (Υπολογισμός αθροίσματος των στοιχείων του πίνακα, υπολογισμός αθροίσματος των θετικών στοιχείων του πίνακα, υπολογισμός μεγαλύτερου στοιχείου του πίνακα, υπολογισμός μικρότερου στοιχείου πίνακα)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #define N 5
4 | int main() {
5 |     int A[N], B[N], i, sum, sumP, max, min;
6 |     for (i=0; i<N; i=i+1) {
7 |         printf("\nGive the element A[%d]= ", i);
8 |         scanf("%d", &A[i]);
```

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία II

```
9      }
10     sum=0;
11     sumP=0;
12     max=A[0];
13     min=A[0];
14     for (i=0; i<N; i=i+1) {
15         sum=sum+A[i];
16         if (A[i]>=0) {
17             sumP=sumP+A[i];
18         }
19         if (A[i]>max) {
20             max=A[i];
21         }
22         if (A[i]<min) {
```

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία III

```
23         min=A[i];
24     }
25 }
26 printf("The Sum is %d \n",sum);
27 printf("The Sum of positive elements is %d \n
    ",sumP);
28 printf("The Max is %d \n",max);
29 printf("The Min is %d \n",min);
30 return 0;
31 }
```

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία IV

Give the element $A[0] = 23$

Give the element $A[1] = 45$

Give the element $A[2] = -4$

Give the element $A[3] = -49$

Give the element $A[4] = 45$

The Max is 45

The Min is -49

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία 2 I

Εκχώρηση τιμών και επεξεργασία στοιχείων μονοδιάστατου πίνακα (Υπολογισμός μεγαλύτερου στοιχείου του πίνακα, υπολογισμός μικρότερου στοιχείου πίνακα - χωρίς αρχικές τιμές)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #define N 5
4 | int main() {
5 |     int A[N], i, max=0, min=0;
6 |     for (i=0; i<N; i=i+1) {
7 |         printf("\nGive the element A[%d]= ", i);
8 |         scanf("%d", &A[i]);
9 |     }
10 |    for (i=0; i<N; i=i+1) {
```

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία 2 II

```
11 |         if (A[i]>max || i==0) {  
12 |             max=A[i];  
13 |         }  
14 |         if (A[i]<min || i==0) {  
15 |             min=A[i];  
16 |         }  
17 |     }  
18 |     printf("The Max is %d \n",max);  
19 |     printf("The Min is %d \n",min);  
20 |     return 0;  
21 | }
```

Μονοδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία 2 III

Give the element $A[0] = 23$

Give the element $A[1] = 45$

Give the element $A[2] = -34$

Give the element $A[3] = -89$

Give the element $A[4] = 87$

The Max is 87

The Min is -89

Contents

- 1 Δομές δεδομένων - Πίνακες
- 2 Αριθμητικοί Πίνακες
 - Μονοδιάστατοι Πίνακες
- 3 Ασκήσεις
- 4 Λύσεις Ασκήσεων

Μονοδιάστατοι Πίνακες - Άσκηση 1

Άσκηση

Να γράψετε ένα πρόγραμμα το οποίο να δημιουργεί και να εμφανίζει τρεις μονοδιάστατους πίνακες με 5 στοιχεία.

- Τον πίνακα A του οποίου τα στοιχεία δίνονται από τον τύπο

$$A[i] = \frac{1}{i + 1}$$

- Τον πίνακα B του οποίου τα στοιχεία θα είναι με αντίστροφη σειρά από τον A
- Τον πίνακα C του οποίου τα στοιχεία θα είναι το άθροισμα των στοιχείων των A και B

Μονοδιάστατοι Πίνακες - Άσκηση 2

Άσκηση

Να γράψετε ένα πρόγραμμα το οποίο να δημιουργεί και να εμφανίζει τους μονοδιάστατους πίνακες A και B με

$$A = [1, 3, 5, \dots, 17, 19]$$

$$B = [20, 18, 16, \dots, 4, 2]$$

και να δημιουργεί και να εμφανίζει τον πίνακα C

$$C = [A, B]$$

- 1 Δομές δεδομένων - Πίνακες
- 2 Αριθμητικοί Πίνακες
 - Μονοδιάστατοι Πίνακες
- 3 Ασκήσεις
- 4 Λύσεις Ασκήσεων

Προτεινόμενη λύση της Άσκησης 1 1

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | #define N 5
4 | int main() {
5 |     int i;
6 |     float A[N], B[N], C[N];
7 |     for (i=0; i<N; i=i+1) {
8 |         A[i]=1/(float) (i+1);
9 |     }
10 |    for (i=0; i<N; i=i+1) {
11 |        B[i]=A[ (N-1) -i];
12 |        C[i]=A[i]+B[i];
13 |    }
14 |    for (i=0; i<N; i=i+1) {
15 |        printf("\t%f", A[i]);
```

Προτεινόμενη λύση της Άσκησης 1 II

```
16     }
17     printf("\n\n");
18     for (i=0;i<N;i=i+1) {
19         printf("\t%f",B[i]);
20     }
21     printf("\n\n");
22     for (i=0;i<N;i=i+1) {
23         printf("\t%f",C[i]);
24     }
25     return 0;
26 }
```

Προτεινόμενη λύση της Άσκησης 2 Ι

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 |     int i,A[10],B[10],C[20];
5 |     for (i=0;i<10;i=i+1) {
6 |         A[i]=2*i+1;
7 |         printf("\t%d\n",A[i]);
8 |     }
9 |     printf("\n\n");
10 |    for (i=0;i<10;i=i+1) {
11 |        B[i]=20-2*i;
12 |        printf("\t%d\n",B[i]);
13 |    }
14 |    printf("\n\n");
15 |    for (i=0;i<20;i=i+1) {
```

Προτεινόμενη λύση της Άσκησης 2 II

```
16     if (i<10){
17         C[i]=A[i];
18     }
19     else{
20         C[i]=B[i-10];
21     }
22     printf("\t%d\n",C[i]);
23 }
24 printf("\n\n");
25 return 0;
26 }
```