



Κεφάλαιο 8.1-8.3

Πίνακες I

(Διάλεξη 15)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Πίνακες (Arrays)



Σε αυτή την ενότητα θα μιλήσουμε για την **δομή δεδομένων Πίνακας**:

1. Εισαγωγή & Σύνταξη
2. Δήλωση Πίνακα
3. Αρχικοποίηση Πίνακα
4. Πρόσβαση
5. Παραδείγματα



1) Πίνακας - Εισαγωγή & Σύνταξη

- Μία **δομή δεδομένων** είναι ένα σύνολο σχετιζομένων τύπων δεδομένων (π.χ. int, char, float,...).
- Ένας **πίνακας** είναι μια απλη δομη δεδομενων στην οποια τα δεδομένα έχουν το ίδιο **όνομα** και τον ίδιο **τύπο**.

π.χ. αντί **int a,b,c;** **→** **int pinakas[3];**

- Οι πίνακες που θα μελετήσουμε σε αυτό το μάθημα είναι **στατικοί**, δηλαδή το μέγεθος τους ορίζεται εκ των προτέρων (πριν την μεταγλώττιση).

Πίνακας Ακεραίων

5	2	6	9	3
0	1	2	3	4

Πίνακας Χαρακτήρων

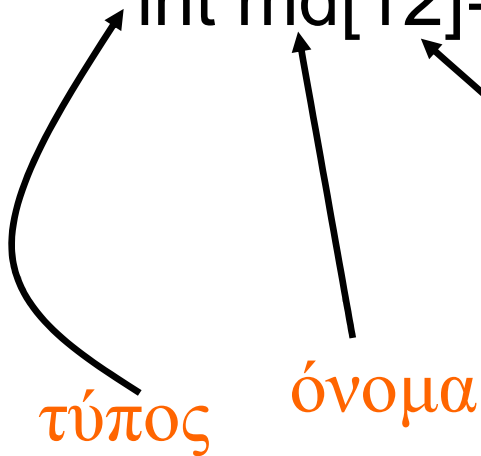
H	E	L	L	O
0	1	2	3	4

1) Πίνακας - Εισαγωγή & Σύνταξη

- Ένας πίνακας είναι μία δομή δεδομένων, όπου ένα σύνολο αντικειμένων του **ιδίου τύπου** αποθηκεύονται **σε σειρά**,

π.χ.

```
int md[12]= {31,28,31,30,31,30,31};
```



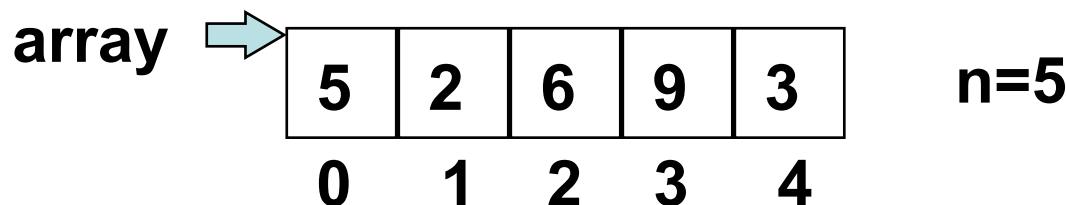
md[0]	31
md[1]	28
md[2]	31
...	
...	
md[11]	31

τιμές



2) Δηλώσεις πινάκων

- `int array [5] ;`
- Η παραπάνω δήλωση δεσμεύει αρκετή μνήμη ώστε να χωρέσουν 5 ακέραιοι.
- **Δεν βάζει** αρχικές τιμές σε αυτές τις θέσεις μνήμης
- Θα δούμε στην συνέχεια τρόπους ανάθεσης τιμών σε αυτές τις θέσεις του πίνακα
π.χ. `int array [5] = { 5, 2, 6, 9, 3 } ;`)
- Τα στοιχεία σε ένα πίνακα με n στοιχεία αριθμούνται από το 0 έως το $(n-1)$. Η **πρώτη θέση** του πίνακα είναι η **θέση 0**



2) Δηλώσεις πινάκων



<τύπος> όνομα[<μέγεθος>;

<τύπος> όνομα[<μέγεθος>]={αρχικοποίηση};

- Το μέγεθος χρειάζεται, εκτός και εάν γίνεται αρχικοποίηση.

π.χ. `char a[3]; // ΣΩΣΤΟ`

π.χ. `char a[]; // ΛΑΘΟΣ`

- **Ορισμός χωρίς μέγεθος**

– το μέγεθος του πίνακα είναι ίσο με τα στοιχεία της αρχικοποίησης

Π.χ. `char pinakas[]={'a', 'b', 'c'}; // ΣΩΣΤΟ`

2) Δηλώσεις Πινάκων



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char a;           // Δήλωση Μεταβλητής
```

```
    char table[5]; // Δήλωση Πίνακα 5 ακεραίων
```

```
    ....
```

```
}
```

2) Δηλώσεις πινάκων (Αποθήκευση)

- Τα στοιχεία ενός πίνακα αποθηκεύονται σε συνεχόμενα κελιά στη μνήμη του υπολογιστή (memory cells).

Πίνακας Ακεραίων



int a[2]={9,5};

Τα δεδομένα είναι συνεχόμενα στην μνήμη

Δήλωση μεταβλητών



int a=9,b=5;

Τα δεδομένα είναι οπουδήποτε στην μνήμη



2) Δηλώσεις πινάκων και χρήση #define

- Συχνά χρησιμοποιούμε #define για να δηλώσουμε το μέγεθος των arrays

```
#define SIZE      39
#define GRADES    5

main ( )
{
    int score [SIZE] ;
    int gradeCount [GRADES] ;
    ...
}
```



2) Δηλώσεις πινάκων

```
int score [39] , gradeCount [6];
```

- Δηλώνει δύο πίνακες τύπου int
- Κανένας πίνακας δεν έχει αρχικοποιηθεί
- Στο score έχει χώρο για 39 στοιχεία (ένα για κάθε μαθητή της τάξης)
- gradeCount έχει χώρο για 6 στοιχεία (ένα για κάθε βαθμό, A-F)

3) Αρχικοποίηση Πίνακα



- Πάντοτε πρέπει να δίνουμε κάποιες αρχικές τιμές στις διάφορες θέσεις του πίνακα.
- Αυτό δεν είναι κάτι νέο. Αυτό κάναμε και με τις μεταβλητές, π.χ.
`int a; a = 0;`
- Μπορούμε να κάνουμε όλες τις θέσεις του πίνακα ίσο με «0», με την πιο κάτω εντολή:
`int a[10]={};`
ή `int a[10]; a[0]=0; a[1]=0;....` **Όχι πρακτικό**

Επίσης, τι γίνεται αν θέλουμε κάποια άλλη τιμή αρχικοποίησης εκτός του 0, θα δούμε στην συνέχεια τον τρόπο...

3) Αρχικοποίηση Πίνακα



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
// Δήλωση & Αρχικοποίηση Πίνακα
```

```
char table[5]={};      ....
```

```
}
```

5) Αρχικοποίηση Πίνακα II



- Πριν είχαμε πει ότι ένας πίνακας αρχικοποιείτε με την εντολή της μορφής **char table[5]={}**;
- Δηλαδή αυτό είναι ισοδύναμο με το πιο κάτω σύνολο εντολών

```
#define MAX 5
```

```
int x[MAX], i;
```

```
for(i=0;i<MAX; i++)
```

```
    x[i]=0;
```

Βλέπουμε ότι το for loop μας επιτρέπει να αρχικοποιήσουμε ένα πίνακα σε ότι τιμή θέλουμε

5) Αρχικοποίηση Πίνακα II



Μπορούμε να αρχικοποιήσουμε τον πίνακα σε
ότι θέλουμε, π.χ.

```
#define SIZE 5
int square[SIZE], i;
/*αρχικοποίηση: στοιχείο i σε i^2*/
for (i = 0; i < SIZE; ++i)
    square[i] = i * i;
```

0	0
1	1
2	4
3	9
4	16

4) Πρόσβαση Πίνακα



- **Σύνταξη: όνομα[αριθμοδείκτης]**
- Ο αριθμοδείκτης πίνακα (subscript ή index) πρέπει να είναι **int**
- $x[0] \Rightarrow$ τιμή πρώτου στοιχείου του πίνακα x
- $x[i] \Rightarrow$ τιμή i -οστού στοιχείου του πίνακα x
 - Το i -οστό στοιχείο βρίσκεται στη θέση $i+1$
- Ευθύνη προγραμματιστή να επαληθεύσει ότι η τιμή αριθμοδείκτη πίνακα είναι στο πεδίο $[0, \text{στοιχεία}-1]$
- Π.χ. `int a[5]={}; a[5]=3;` **ΛΑΘΟΣ** `a[5]` δεν υπάρχει
Προσοχή μόνο το `a[0]` μέχρι `a[4]` υπάρχει

4) Πρόσβαση Πίνακα



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char table[5]={};          // Δήλωση & Αρχικοποίηση Πίνακα
```

```
    table[0] = 'a'; // Αλλαγή θέσης 0 του πίνακα
```

```
    table[1] = 'b'; // Αλλαγή θέσης 1 του πίνακα
```

```
    // Εκτύπωση δυο θέσεων του πίνακα
```

```
    printf("Hello %c %c\n", table[0], table[1]);
```

```
    // Αλλαγή τιμών του πίνακα με τιμές εισόδου από το πληκτρολόγιο
```

```
    scanf("%c %c", &table[0], &table[1]);
```

```
    printf("Hello %c %c\n", table[0], table[1]);
```

```
}
```

ΤΥΠΩΝΕΙ:

“Hello a b \n Hello input1 input2”

4) Πρόσβαση Πίνακα



```
int x[]={1,2,3,5,7,11,13,17,19,23,29,31};
```

Πως μοιάζει ο πίνακας μετά την εκτέλεση των πιο κάτω εντολών?

```
x[3] = 8258;
```

```
sum = x[0] + x[1];
```

```
x[3] += 1;
```

```
x[2] = x[0] + x[1];
```

Ο πίνακας τώρα είναι ως εξής :

```
int x[]={1,2,3,8259,7,11,13,17,19,23,29,31};
```

4) Πρόσβαση Πίνακα



Όλες οι πιο κάτω εντολές είναι ορθές

```
float f = 5.0;
```

```
x[i] = 0;
```

```
x[i] = x[j];
```

```
x[j+k*4]= x[u] + 3;
```

```
x[x[1]]= 11;
```

```
diff = x[y]-x[foo()];
```

```
total =total + x[i++];
```

```
(x[i]== x[(int)f])
```

Πίνακας Ακεραίων

x:

5	2	6	9	3
0	1	2	3	4

int foo()

δηλαδή:

total=total+x[i];

i++;



5) Παράδειγμα βαθμών

Γράψετε ένα πρόγραμμα το οποίο θα δέχεται από τον χρήστη τις βαθμολογίες 39 φοιτητών από το πληκτρολόγιο θα τις αποθηκεύει σε πίνακα και στην συνέχεια θα εκτυπώνει τον μέσο όρο, και πόσα A,B,C,D,E,F υπάρχουν.

```
#include <stdio.h>
#define SIZE      39
#define GRADES    6

main ( )
{
    int i, total, score [SIZE] ;
    int gradeCount [GRADES]={} ;
    float average;
```

Συνέχεια...

```
/* Fill score array with scores */
for ( i = 0 ; i < SIZE ; i++)
{
    printf ( "Enter next score : " ) ;
    scanf ( "%d ", &score [ i ] ) ;
}
```

90	<=	A	<=	100
80	<=	B	<	90
70	<=	C	<	80
60	<=	D	<	70
50	<=	E	<	60
0	<=	F	<	50



5) Παράδειγμα βαθμών (συνέχεια)

```
/* Calculate total & count grades */
for ( i = 0 ; i < SIZE ; i++)
{
    total += score [ i ] ;
    switch ( score [ i ] / 10 )
    {
        case 10 :
        case 9 : gradeCount [5]++ ;
                 break ;
        case 8 : gradeCount [4]++ ;
                 break ;
        case 7 : gradeCount [3]++ ;
                 break ;
        case 6 : gradeCount [2]++ ;
                 break ;
        case 5 : gradeCount [1]++ ;
                 break ;
        default : gradeCount [0]++ ;
    }
}
```

↑ **Συνέχεια...**

```
average = (float) total / SIZE ;
```

```
/* Print results */
```

```
printf ( "The class average is %.2f\n",
         average ) ;
```

```
printf ( "%2d As\n", gradeCount [5] ) ;
printf ( "%2d Bs\n", gradeCount [4] ) ;
printf ( "%2d Cs\n", gradeCount [3] ) ;
printf ( "%2d Ds\n", gradeCount [2] ) ;
printf ( "%2d Es\n", gradeCount [1] ) ;
printf ( "%2d Fs\n", gradeCount [0] ) ;
```