Κεφάλαιο 8.7

Πολυδιάστατοι Πίνακες

(Διάλεξη 18)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Πολυδιάστατοι πίνακες

- Μέχρι τώρα μιλούσαμε για **Μονοδιάστατους Πίνακες.**

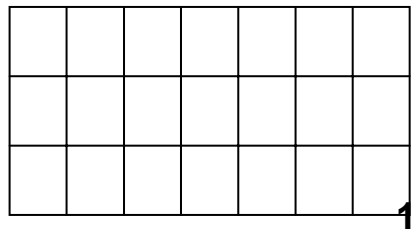
  π.χ. int age[5]= {31,28,31,30,31};

ή

για **Παράλληλους πίνακες,** π.χ.

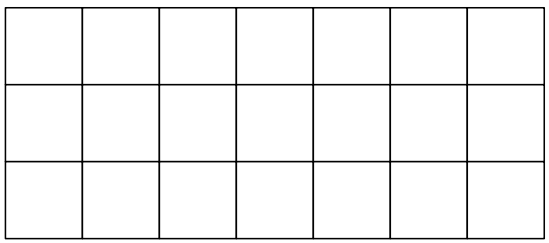
```
int id[5]    =  {1029,1132,1031,9991,1513};
int age[5] =  {31,28,31,30,31};
```

- **Σήμερα θα μιλήσουμε για Πολυδιάστατους Πίνακες (κυρίως Δισδιάστατους).**

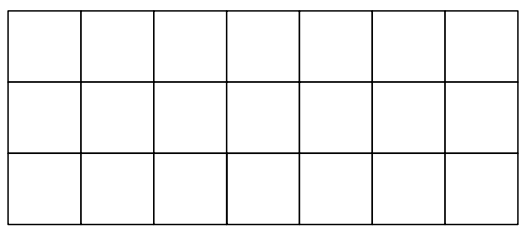


Πολυδιάστατοι πίνακες

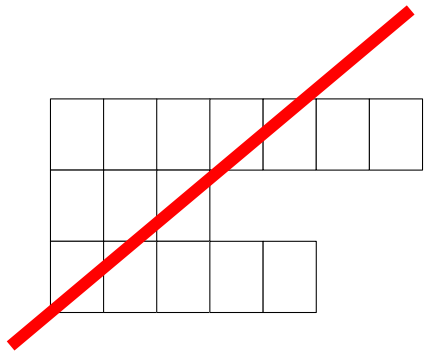
- Η C διαθέτει ορθογωνικούς πολυδιάστατους πίνακες.
  δηλ. **char array[3][7]** /*[γραμμή][στήλη]*/



- Αυτό συνεπάγεται ότι δεσμεύουμε ακριβός 3*7 = 21 χαρακτήρες στην μνήμη. Κάθε χαρακτήρας είναι 1 byte.
- Επομένως η πιο πάνω δήλωση **δεσμεύει 21 bytes.**
- Για να αναφερθούμε σε κάποιο στοιχείο του πίνακα χρησιμοποιούμε δείκτες θέσης.
  π.χ. **array[0][0]='A', array[2][3]='B'**
  ~~**array[3][2]='A'**~~ /*[γραμμή][στήλη]*/
  Δέν υπάρχει το [3][2]

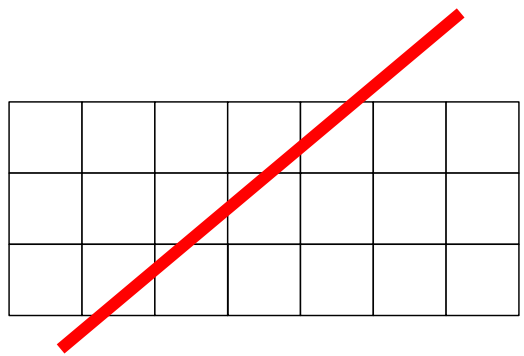


Κοινά Λάθη σε Πολυδιάστατους Πίνακες



Ερώτηση Α

- Μπορούμε να δηλώσουμε **μη-ορθογωνικούς** πολυδιάστατους πίνακες? **ΟΧΙ**
- Θα μπορούσε να γίνει κάτι αντίστοιχο, με χρήση δυναμικής δέσμευσης μνήμης, αλλά όχι σε αυτό το μάθημα.



Ερώτηση Β

- Μπορούν τα στοιχεία του πίνακα να είναι με να είναι **ανομοιογενή (**δηλαδή κάποιες γραμμές η στήλες να φέρουν διαφορετικό τύπο (int, char, float)? **ΟΧΙ**
- Θα μπορούσε να γίνει κάτι αντίστοιχο, με την χρήση κάποιων ειδικών δομών που λέγονται structs, αλλά όχι σε αυτό το μάθημα.

> 2-διάστατοι Πίνακες

- Ένας πολυδιάστατος πίνακας είναι ένας **μονοδιάστατος πίνακας** κάθε στοιχείο του οποίου **είναι ένας πίνακας**.
- Μπορούν να υπάρχουν πίνακες σε πολλές διαστάσεις (η **ANSI-C** που χρησιμοποιούμε υποδεικνύει στους κατασκευαστές μεταγλωττιστών να προσφέρουν **μέχρι 6 διαστάσεις**)
- Σε αυτό το μάθημα θα μελετήσουμε **δισδιάστατους πίνακες** (η άλλη χρήσιμη κατηγόρια πινάκων είναι οι τρισδιάστατοι).
- **Παράδειγμα 3-διάστατου πίνακα**

  int enrolled[course][campus][year];

  Π.χ. εγγραφές της μορφής

  {32,1,2004}, {132,1,2005},{131,1,2005}

Πολυδιάστατοι πίνακες (συν.)

- **Αρχικοποίηση 2-διάστατου πίνακα Στατικά**

  Μια λίστα αρχικών τιμών κλεισμένη σε άγκιστρα, όπου κάθε τιμή παίρνει αρχική τιμή από μια αντίστοιχη υπολίστα, π.χ.

```
int studentGrades[100][6] = {
        {99, 76, 88 , 74,  65, 53 },
        {67, 71, 77 , 71,  80, 47 },
};
```

Η πρώτες δυο γραμμές (από τις 100) περιέχουν τις 6 βαθμολογίες δυο φοιτητών)

- **Αρχικοποίηση 2-διάστατου πίνακα με FOR loop**

```
for(i=0;i<100;i++)                  // γραμμές
   for(j=0;j<6; j++)                // στήλες
     studentGrades[i][j]=0;
```

- **Το nesting-level είναι ίδιο με τον αριθμό των διαστάσεων του πίνακα**

Παράδειγμα

Γράψετε ένα πρόγραμμα το οποίο
α) δημιουργεί ένα πίνακα 10x10

β) Αρχικοποιεί κάθε θέση του πίνακα σε 0

γ) Εκτυπώνει τον πίνακα

Παράδειγμα 1 – Λύση Α

```
#include <stdio.h>
#define SIZE 10

main () {
    // Δήλωση Πίνακα
    int matrix[SIZE][SIZE], i, j;

    // Αρχικοποίηση πίνακα
    for (i=0; i<SIZE; i++)
        for (j=0; j<SIZE; j++)
            matrix[i][j] = 0;

    // Εκτύπωση πίνακα
    for (i=0; i<SIZE; i++) {
        for (j=0; j<SIZE; j++) {
            printf("%d", matrix[i][j]);
        }
        printf("\n");
    }

}
```

Εκτυπώνει

```
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
```

Μετά την εκτύπωση κάθε γραμμής εκτυπώνουμε newline

Παράδειγμα 1 – Λύση Β

```
#include <stdio.h>

#define SIZE 10

main () {

    int matrix[SIZE][SIZE]  = {{ }};
    int i,j;

    // Εκτύπωση πίνακα
    for (i=0; i<SIZE; i++) {

        for (j=0; j<SIZE; j++) {
            printf("%d", matrix[i][j]);
        }
        printf("\n");
    }

}
```

Η Αρχικοποίηση γίνεται εδώ με δυο παρενθέσεις αντί for loop.

Το μειονέκτημα είναι ότι δουλεύει μόνο για αρχικοποίηση σε 0

Παράδειγμα 2
Αρχικοποίηση Πίνακα

Γράψετε ένα πρόγραμμα το οποίο δημιουργεί τον πιο κάτω 2-διάστατο πίνακα στην μνήμη και στην συνέχεια τον εκτυπώνει

34567
34567
34567
34567
34567

Παράδειγμα 2
Αρχικοποίηση Πίνακα

Βλέπουμε ότι ο πίνακας Α μπορεί να παραχθεί από τον Β εάν προσθέσουμε σε κάθε θέση του Β το αριθμό 3

A	B
34567	01234
34567	01234
34567	01234
34567	01234
34567	01234

Παράδειγμα 2
Αρχικοποίηση Πίνακα

```
#include <stdio.h>

#define SIZE 5

main () {

        int matrix[SIZE][SIZE];
        int i,j;
        // Αρχικοποιηση
        for (i=0; i<SIZE; i++)
                for (j=0; j<SIZE; j++)
                        matrix[i][j] = j+3;

        for (i=0; i<SIZE; i++) {
                for (j=0; j<SIZE; j++) {
                        printf("%d", matrix[i][j]);
                }
                printf("\n");
        }
}
```

Για κάθε i (γραμμή) εκτελείτε το πιο κάτω:

```
j
0+3=3
1+3=4
2+3=5
3+3=6
4+3=7
```

Στο τέλος εκτυπώνετε

```
34567
34567
34567
34567
34567
```

Παράδειγμα 3
Άθροισμα Διαγωνίου

Γράψετε ένα πρόγραμμα το οποίο δημιουργεί τον πιο κάτω 2-διάστατο πίνακα στην μνήμη και στην συνέχεια εκτυπώνει το άθροισμα της διαγωνίου

34567
34567
34567 **= 25**
34567
34567

Παρατηρούμε ότι οι τιμές που θέλουμε είναι για i=j

Παράδειγμα 3
Άθροισμα Διαγωνίου

```
#include <stdio.h>
#define SIZE 5
main () {
        int matrix[SIZE][SIZE]; int i,j;
        int sum = 0;
        // Αρχικοποίηση
        for (i=0; i<SIZE; i++)
                for (j=0; j<SIZE; j++)
                        matrix[i][j] = j+3;

        // Εύρεση Αθροίσματος
        for (i=0; i<SIZE; i++) {
                for (j=0; j<SIZE; j++) {
                        if (i==j) {
                            sum += matrix[i][j];
                        }
                }
        }
        printf("Sum: %d", sum);
}
```

Μεταβλητή που θα κρατά το άθροισμα

```
for (i=0; i<SIZE; i++) {
    sum += matrix[i][i];

}
```

Παράδειγμα 4
Άθροισμα Στήλης

Γράψετε ένα πρόγραμμα το οποίο δημιουργεί τον πιο κάτω 2-διάστατο πίνακα στην μνήμη και στην συνέχεια εκτυπώνει το άθροισμα της τρίτης στήλης (index 2)

34567
34567
34567
34567
34567

= 25

Παρατηρούμε ότι θέλουμε τις τιμές όπου το j=2

Παράδειγμα 3
Άθροισμα Στήλης

```
#include <stdio.h>
#define SIZE 5
main () {
        int matrix[SIZE][SIZE]; int i,j;
        int sum = 0;
        // Αρχικοποίηση
        for (i=0; i<SIZE; i++)
             for (j=0; j<SIZE; j++)
                        matrix[i][j] = j+3;

        // Εύρεση Αθροίσματος
        for (i=0; i<SIZE; i++) {
              sum += matrix[i][2];
        }
        printf("Sum: %d", sum);
}
```

Εδώ το δεύτερο for loop θα ήταν αχρείαστο.

Πολυδιάστατοι Πίνακες & Συναρτήσεις

- Όταν περνάμε **πολυδιάστατους πίνακες** σε συναρτήσεις, τότε ακολουθείτε η ίδια λογική με την περίπτωση των **μονοδιάστατων πινάκων**

- Δηλαδή οι πίνακες περνάνε **δια αναφοράς** (και όχι δια τιμής).

- Επομένως **δεν παράγεται ένα νέο αντίγραφο** του πίνακα αλλά αντίθετα, η συνάρτηση μπορεί να κάνει κατευθείαν αλλαγές πάνω στον αρχικό πίνακα.

Πίνακες και Συναρτήσεις (Πρότυπο – Ορισμός - Κλήση)

```
#include <stdio.h>
#define SIZE 4
// Πρότυπο
void  FillArray (int[ ][ ]) ;

main ( ) {
    int array [SIZE][SIZE];

    // Κλήση Συνάρτησης
    FillArray (array);
}
```

```
// Ορισμός Συνάρτησης
void FillArray( int table[ ][SIZE] )
{
      int i,j;

      for ( i = 0; i < SIZE; i++)    {
            for ( j = 0; j < SIZE; j++)
            {       table[i][j] = 0;   }
       }
 }
```

Μόνο η πρώτη διάσταση μπορεί να παραληφθεί

Έξοδος:

array[0][1] = 0	array[0] [1] = 0	array[0] [2] = 0	array[0] [3] = 0
array[1][1] = 0	array[1] [1] = 0	array[1] [2] = 0	array[1] [3] = 0
array[2][1] = 0	array[2] [1] = 0	array[2] [2] = 0	array[2] [3] = 0
array[3][1] = 0	array[3] [1] = 0	array[3] [2] = 0	array[3] [3] = 0

Παράδειγμα Επεξεργασίας Βαθμών

- Γράψετε ένα πρόγραμμα:
  - που διαβάζει από το πληκτρολόγιο βαθμούς 80 φοιτητών. Για κάθε φοιτητή διαβάζονται 5 βαθμοί δηλαδή:

    90 60 70 89 45 (βαθμοί 1ου φοιτητή)
    40 56 78 99 100 (βαθμοί 2ου φοιτητή)
    κτλ
  - υπολογίζει και τυπώνει

    (α) βαθμούς κάθε φοιτητή

    (β) τον μέσο όρο για κάθε φοιτητή

Μέσος Όρος

90.00 60.00 70.00 89.00 45.00 70.80
40.00 56.00 78.00 99.00 100.00 74.60

Παράδειγμα Επεξεργασίας Βαθμών

Τι πρέπει να γίνει;

Α) Διαβαστούν δεδομένα από το πληκτρολόγιο και να αποθηκευτούν σε ένα 2-διαστάσεων πίνακα

Β) Υπολογισμός μέσου όρου ανά φοιτητή

C) Εκτύπωση αποτελεσμάτων

Παράδειγμα Επεξεργασίας Βαθμών

// Σταθερές

```
#define NUM_STUDENTS 80
#define NUM_COURSES    5
```

Πίνακες στο main

```
// Πίνακας ο οποίος κρατά για κάθε φοιτητή την βαθμολογία
// για κάθε μάθημα
float grade_table[NUM_STUDENTS][NUM_COURSES] = {{ }};

// Πίνακας ο οποίος κρατά για κάθε φοιτητή τον
// μέσο όρο της βαθμολογίας
float average_per_student[NUM_STUDENTS]={ };
```

Παράδειγμα Επεξεργασίας Βαθμών

```
main()
{
        float grade_table[NUM_STUDENTS][NUM_COURSES] = {{ }};
        float average_per_student[NUM_STUDENTS]={ };

        // Διάβασμα δεδομένων (αποθηκευτούν)
        read_data(grade_table);

       // Υπολογισμός μέσων όρων
    compute_averages(grade_table, average_per_student);

        // Εκτύπωση αποτελεσμάτων
        display_results(grade_table, average_per_student);
}
```

Παράδειγμα Επεξεργασίας Βαθμών

```
void read_data(float grade_table[][NUM_COURSES])
{
        int i,j;

        for(i=0;i<NUM_STUDENTS;i++) {
                printf("Student %d\n", i);
                for(j=0;j<NUM_COURSES;j++) {
                        scanf("%f", &grade_table[i][j]);
                }
        }
}
```

Παράδειγμα Επεξεργασίας Βαθμών

Δηλώνουμε ότι απαγορεύετε να αλλάξει ο πίνακας σε αυτή την συνάρτηση (π.χ. από κάποιο προγραμματιστικό λάθος)

```
void compute_averages(
        const float grade_table[][NUM_COURSES],
        float average_per_student[])
{
        int i,j;
        for(i=0;i<NUM_STUDENTS; i++) {
                for(j=0;j<NUM_COURSES; j++){
                        average_per_student[i]+=grade_table[i][j];
                }
        }

        for(i=0;i<NUM_STUDENTS;++i)
                average_per_student[i]/=NUM_COURSES;
}
```

Κρατά το Sum κάθε φοιτητή

Παράδειγμα Επεξεργασίας Βαθμών

```
void display_results(float grade_table[][NUM_COURSES],
                     float average_per_student[])
{
        int i,j;

        for(i=0;i<NUM_STUDENTS; i++){
                for(j=0;j<NUM_COURSES;j++) {
                        printf("%5.2f  ",grade_table[i][j]);
                }
                printf("%5.2f\n",average_per_student[i]);
        }
}
```

Μέσος Όρος

Εκτυπώνει:

```
90.00  60.00  70.00  89.00  45.00   70.80
40.00  56.00  78.00  99.00  100.00  74.60
```