

ΠΙΝΑΚΕΣ

ΟΡΙΣΜΟΣ

Πίνακας (array) είναι ένα σύνολο από θέσεις μνήμης (μεταβλητές), όπου η κάθε μία περιέχει τον ίδιο τύπο δεδομένων (π.χ int, float κλπ). Κάθε θέση ονομάζεται στοιχείο του πίνακα. Δηλώνουμε έναν πίνακα στο πρόγραμμα γράφοντας τον τύπο, το όνομα του πίνακα και τέλος έναν δείκτη (subscript) μέσα σε τετράγωνες αγκύλες που είναι ο αριθμός των στοιχείων του πίνακα.

Δηλαδή:

```
double MyArray[10];
```

όπου:

- double: τύπος
- MyArray: όνομα πίνακα
- [10]: δείκτης πλήθους στοιχείων πίνακα

ΜΕΓΕΘΟΣ ΣΤΗ ΜΝΗΜΗ

Ο μεταγλωττιστής για αυτόν τον πίνακα δεσμεύει 80 bytes (δηλ. double 8 bytes x 10 στοιχεία).

Τα στοιχεία του πίνακα μετριοούνται από το 0, δηλαδή το πρώτο στοιχείο του πίνακα είναι MyArray[0], το δεύτερο MyArray[1], το τρίτο MyArray[2] κλπ.

Άρα εάν έχουμε έναν πίνακα με **n** στοιχεία, η αρίθμησή τους είναι:

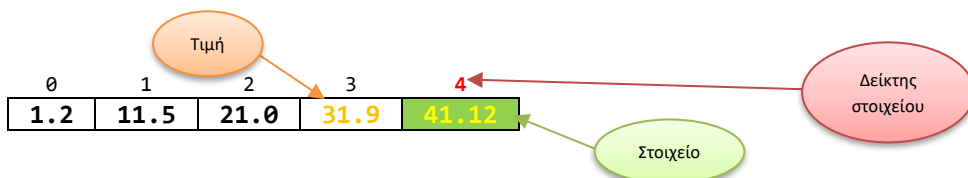
MyArray[0], MyArray[1], MyArray[n-1]

Οπότε ο πίνακας του παραδείγματος έχει στοιχεία που μετριοούνται από το MyArray[0] έως το MyArray[9].

ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΟΝΟΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ

Αρχικοποιούμε έναν μονοδιάστατο πίνακα γράφοντας την δήλωση:

```
float MyArray[5] = {1.2, 11.5, 21.0, 31.9, 41.12};
```



Αυτό σημαίνει ότι το στοιχείο:

- MyArray[0] θα πάρει την τιμή 1.2,
- το MyArray[1] την τιμή 11.5,
- το MyArray[2] την τιμή 21.0,
- το MyArray[3] την τιμή 31.9,
- και το MyArray[4] την τιμή 41.12.

ΠΑΡΑΤΗΡΗΣΕΙΣ

1. Εάν στη δήλωση παραλείψουμε τον δείκτη του πίνακα αλλά βάλουμε τις τιμές, τότε ο compiler θα φτιάξει έναν πίνακα που να χωρέσει τις τιμές αυτές.
2. Δεν πρέπει να αρχικοποιούμε έναν πίνακα με τιμές περισσότερες από το μέγεθός του.
3. Εάν αρχικοποιήσουμε έναν πίνακα με λιγότερες τιμές τότε θα αρχικοποιηθούν τα πρώτα στοιχεία με τις δοθείσες τιμές και τα υπόλοιπα με τιμές 0.

Προσοχή στον δείκτη μέσα στις αγκύλες.

Στη δήλωση:

```
float MyArray[5];
```

ο δείκτης 5 καθορίζει το μέγεθος του πίνακα,

ενώ στη δήλωση:

```
MyArray[4]=41.12;
```

ο δείκτης 4 δηλώνει ότι το 5ο στοιχείο του πίνακα που αναφερόμαστε, θα πάρει την τιμή 41.12.

ΔΙΣΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

Δισδιάστατοι είναι οι πίνακες των οποίων τα στοιχεία είναι επίσης πίνακες.

Η πρόταση :

```
int Myarray[2][4];
```

δηλώνει το Myarray σαν πίνακα ακεραίων 2 στοιχείων, που κάθε στοιχείο του είναι ένας πίνακας ακεραίων 4 στοιχείων.

Στην C και την C++ δεν υπάρχει περιορισμός στον αριθμό διαστάσεων ενός πίνακα.

Για να αναφερθούμε σε ένα στοιχείο ενός δισδιάστατου πίνακα θα πρέπει να καθοριστούν σωστά οι δείκτες. Δηλ. η έκφραση:

```
Myarray[1]
```

αναφέρεται στην δεύτερη γραμμή του πίνακα Myarray.

Ενώ η έκφραση:

```
Myarray[0][3]
```

αναφέρεται στο τέταρτο στοιχείο της πρώτης γραμμής του πίνακα Myarray.

Άρα οι δείκτες χρησιμοποιούνται για να δείξουν:

```
Myarray[γραμμή][στοιχείο]
```

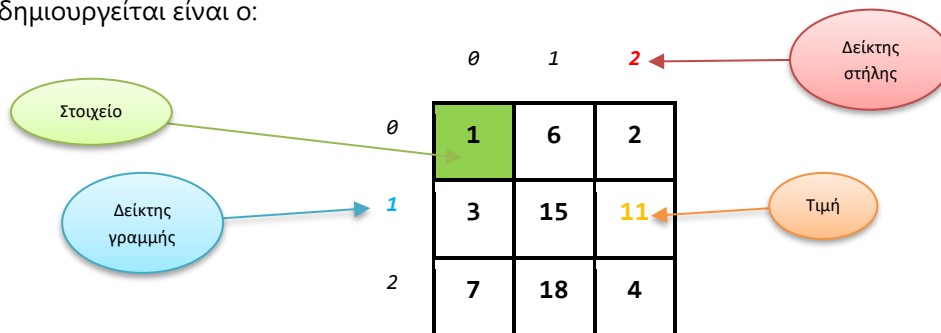
ΑΡΧΙΚΟΠΟΙΗΣΗ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ

Για να αρχικοποιηθεί ένας δισδιάστατος πίνακας, θα πρέπει κάθε γραμμή με τιμές, να περικλείεται από άγκιστρα. Αν δεν υπάρχουν σε κάποια στοιχεία τιμές τότε τα στοιχεία αυτά παίρνουν την τιμή 0.

Όταν εκτελεστεί η πρόταση:

```
int array[3][3]={ {1,6,2},{3,15,11},{7,18,4} };
```

ο πίνακας που δημιουργείται είναι ο:



ΑΝΑΓΝΩΡΙΣΤΙΚΟ define

Στην C++ χρησιμοποιούμε το αναγνωριστικό **#define** για να αντικαταστήσουμε μία έκφραση, μία δήλωση, ή έναν αριθμό που δηλώνει στοιχεία ενός πίνακα με ένα γράμμα ή μία λέξη, έτσι ώστε να μην χρειάζεται να θυμόμαστε τον αριθμό των στοιχείων.

Η σύνταξη είναι η εξής:

```
# define <αναγνωριστικό> <αντικατατάτης>
```

Π.χ τον αριθμό των στοιχείων ενός μονοδιάστατου πίνακα μπορούμε σε όλο το πρόγραμμα να τον αντικαταστήσουμε με ένα γράμμα (π.χ N) ή μια λέξη (π.χ. CITY).

```
# define N 10 ή # define CITY 10
```

ΧΡΗΣΗ ΒΡΟΧΟΥ FOR ΓΙΑ ΔΗΛΩΣΗ ΚΑΙ ΧΕΙΡΙΣΜΟ ΠΙΝΑΚΩΝ

Για τη αρχικοποίηση κα χρήση των πινάκων και δεδομένου ότι έχουμε συγκεκριμένο αριθμό στοιχείων να διαχειριστούμε, χρησιμοποιούμε την δομή επανάληψης *for* η οποία μας δίνει τη δυνατότητα να προσπελάσουμε όλα τα στοιχεία ενός πίνακα με τη σειρά είτε αυτός είναι μονοδιάστατος όπου χρησιμοποιούμε έναν βρόχο ή δισδιάστατος όπου χρησιμοποιούμε δύο βρόχους τον έναν εμφωλευμένο μέσα στον άλλο.

ΠΑΡΑΔΕΙΓΜΑΤΑ FOR ΓΙΑ ΜΟΝΟΔΙΑΣΤΑΤΟΥΣ ΠΙΝΑΚΕΣ ΜΕ ΕΝΑΝ ΒΡΟΧΟ

Διάβαση στοιχείων ενός μονοδιάστατου πίνακα N στοιχείων

```
for (int i = 0; i < N; i++)
    cin >> arr[i];
```

Εμφάνιση στοιχείων ενός μονοδιάστατου πίνακα N στοιχείων

```
for (int i = 0; i < N; i++)
    cout << arr[i];
```

Άθροισμα στοιχείων ενός μονοδιάστατου πίνακα N στοιχείων με τη χρήση μεταβλητής αθροίσματος *sum*

```
int sum = 0;
for (int i = 0; i < N; i++)
    sum = sum + arr[i];
```

ΠΑΡΑΔΕΙΓΜΑΤΑ FOR ΓΙΑ ΔΙΣΔΙΑΣΤΑΤΟΥΣ ΠΙΝΑΚΕΣ ΜΕ ΔΥΟ ΒΡΟΧΟΥΣ

Διάβαση στοιχείων ενός δισδιάστατου πίνακα M γραμμών και N στηλών.

```
for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
        cin >> arr[i][j] ;
```

Εμφάνιση στοιχείων ενός δισδιάστατου πίνακα M γραμμών και N στηλών.

```
for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
        cout << arr[i][j];
```

Άθροισμα στοιχείων ενός δισδιάστατου πίνακα M γραμμών και N στηλών με τη χρήση μεταβλητής αθροίσματος.

```
int sum = 0;
for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
        sum = sum + arr[i][j];
```

Άθροισμα στοιχείων ανά γραμμή ενός δισδιάστατου πίνακα M γραμμών και N στηλών με τη χρήση βοηθητικού μονοδιάστατου πίνακα.

```
for (int i = 0; i < M; i++){
    row[i] = 0;
    for (int j = 0; j < N; j++)
        row[i] = row[i] + arr[i][j];
}
```

Άθροισμα στοιχείων ανά στήλη ενός δισδιάστατου πίνακα M γραμμών και N στηλών με τη χρήση βοηθητικού μονοδιάστατου πίνακα.

```
for (int j = 0; j < N; j++){
    col[j] = 0;
    for (int i = 0; i < M; i++)
        col[j] = col[j] + arr[i][j];
}
```

ΠΑΡΑΔΕΙΓΜΑΤΑ ΔΗΛΩΣΗΣ ΚΑΙ ΧΡΗΣΗΣ ΠΙΝΑΚΩΝ

Παράδειγμα 1: ΔΗΛΩΣΗ ΚΑΙ ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΟΝΟΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ

Το παρακάτω πρόγραμμα αρχικοποιεί έναν ακέραιο πίνακα με τιμές που του δίνει ο χρήστης.

```
#include <iostream>
#define Line 5 //όπου line αντικατέστησέ το με 5
using namespace std;

int main(){

    int oneDimArray[Line]; //Δήλωση πίνακα ακεραίων πλήθους Line
    int i; //Δήλωση μετρητή των βρόχων

    /* Βρόχος για την εισαγωγή των τιμών
    του πίνακα από τον χρήστη */
    for(i = 0; i < Line; i++){
        cout << "Give value for myArray[" << i << "]: ";
        cin >> oneDimArray[i];
    }

    cout << endl;

    // Βρόχος για την εκτύπωση των τιμών του πίνακα
    for(i = 0; i < Line; i++){
        cout << "myArray[" << i << "] has the Value: ";
        cout << oneDimArray[i] << endl;
    }

    return 0;
}
```

Παράδειγμα 2: ΧΕΙΡΙΣΜΟΣ ΔΕΔΟΜΕΝΩΝ ΜΟΝΟΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ

Το παρακάτω πρόγραμμα ζητά από τον χρήστη να αρχικοποιήσει με τιμές έναν πίνακα. Κατόπιν να διπλασιάζει τις τιμές του πίνακα και να τις εκτυπώνει με στηλοθεσία (tab).

```
#include <iostream>
#define Row 5
using namespace std;

int main(){
    int oneDimArray[Row], i;

    /* Βρόχος για την εισαγωγή των τιμών
    του πίνακα από τον χρήστη */
    for(i = 0; i < Row; i++){
        cout << "array[" <<i << "]: ";
        cin >> oneDimArray[i];
    }

    cout<<endl;

    /* Βρόχος για τον διπλασιασμό
    των τιμών του πίνακα */
    for(i = 0; i < Row; i++)
        oneDimArray[i] *= 2;

    /* Βρόχος για την εκτύπωση των νέων τιμών του πίνακα
    με στηλοθεσία (με ένα tab ανάμεσά τους*/
    for(i = 0; i < Row; i++)
        cout << oneDimArray[i] << "\t";
}
```

```
return 0;
}
```

Παράδειγμα 3: ΔΗΛΩΣΗ ΚΑΙ ΑΡΧΙΚΟΠΟΙΗΣΗ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ

Το παρακάτω πρόγραμμα αρχικοποιεί έναν ακέραιο δισδιάστατο πίνακα 3X3 με τιμές που του δίνει ο χρήστης.

```
#include <iostream>
#define Row 3 //Καθορισμός γραμμών πίνακα
#define Column 3 //Καθορισμός στηλών πίνακα
using namespace std;

int main(){
    //Δήλωση δισδιάστατου πίνακα ακεραίων 3X3
    int twoDimArray[Row][Column];
    int i, j; //Δήλωση μετρητών των βρόχων

    /* Διπλός βρόχος για την εισαγωγή των τιμών
    του πίνακα από τον χρήστη */
    for(i = 0; i < Row; i++){
        for(j = 0; j < Column; j++){
            cout << "Give value for myArray[" << i << "][" << j << "]: ";
            cin >> twoDimArray[i][j];
        }

        cout << endl;

    // Διπλός βρόχος για την εκτύπωση των τιμών του πίνακα
    for(i = 0; i < Row; i++){
        for(j = 0; j < Column; j++){
            cout << "myArray[" << i << "][" << j << "] has value: ";
            cout << twoDimArray[i][j] << endl;
        }

        return 0;
    }
}
```