

Semana #4 ARREGLOS BIDIMENSIONALES (Matrices)

Un arreglo bidimensional es lo que se conoce en otros lenguajes de programación e inclusive en el mismo C++ dependiendo del enfoque que se le dé como **una matriz**.

La matriz se compone de filas y columnas y se declara de la siguiente manera:

```
int n[5][5]
```

El primer valor llevara control de las filas y el segundo valor de las columnas. Para hacer el recorrido como han visto en el ejemplo anterior se utilizan dos ciclos for, uno que controla las filas (variable i) y otro que recorre las columnas (variable j). En este caso, el primer ciclo esperara a que se ejecute por completo el for que se encuentra dentro de él, para luego pasar a la siguiente fila.

```
/*1. Hacer un programa para rellenar una matriz pidiendo al usuario el número de filas
y columnas, Posteriormente mostrar la matriz en pantalla.*/

#include<iostream>
#include<stdlib.h>

using namespace std;

int main() {
    int numeros[100][100], filas, columnas;

    cout << "Digite el numero de filas: "; cin >> filas;
    cout << "Digite el numero de columnas: "; cin >> columnas;

    //Rellenando la matriz
    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            cout << "Digite un numero [" << i << "][" << j << "]: ";
            cin >> numeros[i][j];
        }
    }

    cout << "\nMostrando matriz\n\n";

    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            cout << numeros[i][j];
        }
        cout << "\n";
    }
    system("pause");
    return 0;
}
```

```
/*2. Realiza un programa que defina una matriz de 3x3 y escriba un ciclo para que
muestre la diagonal principal de la matriz.*/

#include<iostream>
#include<stdlib.h>

using namespace std;

int main() {
    int numeros[3][3] = { {1,2,3},{4,5,6},{7,8,9} };

    cout << "Mostrando Matriz completa\n";

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << numeros[i][j];
        }
        cout << "\n";
    }

    cout << "\n\nMostrando diagonal principal de la matriz\n";

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (i == j) {
                cout << numeros[i][j] << endl;
            }
        }
    }

    system("pause");
    return 0;
}
```

```
/*3. Hacer una matriz de tipo entera de 2 * 2, llenarla de números y luego copiar todo
su contenido hacia otra matriz.*/

#include<iostream>
#include<stdlib.h>
using namespace std;

int main() {
    int matriz1[2][2] = { {1,2},{3,4} };
    int matriz2[2][2];

    //Copiando el contenido de la matriz1 hacia matriz2
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            matriz2[i][j] = matriz1[i][j];
        }
    }

    //Mostrando matriz2
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            cout << matriz2[i][j];
        }
        cout << "\n";
    }

    system("pause");
    return 0;
}
```

Función rand y srand

La función **rand ()** se usa en C para generar números aleatorios. Si generamos una secuencia de número aleatorio con la función rand (), creará la misma secuencia una y otra vez cada vez que se ejecute el programa. Digamos que si estamos generando 5 números aleatorios en C con la ayuda de rand () en un bucle, cada vez que compilamos y ejecutamos el programa, nuestra salida debe ser la misma secuencia de números

La función **srand ()** establece la raíz que utiliza rand para generar números "aleatorios".

Si no llamas a srand antes de tu primera llamada a rand, es como si hubieras llamado a srand (1) para establecer la raíz en uno. En resumen, srand (): establece la raíz para la función rand ()

srand (tiempo (NULL)); utiliza el reloj interno de la computadora para controlar la elección de la raíz.

Veamos un ejemplo:

```
/*4. Hacer una matriz preguntando al usuario el numero de filas y columnas, llenarla de
números aleatorios, copiar el contenido a otra matriz y por último mostrarla en
pantalla.*/

#include<iostream>
#include<stdlib.h>
#include<time.h>

using namespace std;

int main() {
    int matriz[100][100], filas, columnas;
    int matriz2[100][100], dato = 0;

    cout << "Digite el numero de filas: "; cin >> filas;
    cout << "Digite el numero de columnas: "; cin >> columnas;

    srand(time(NULL));

    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            dato = 1 + rand() % (100); //números aleatorios de 1 a 100
            matriz[i][j] = dato;
        }
    }

    //Copiando el contenido a una nueva matriz
    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            matriz2[i][j] = matriz[i][j];
        }
    }

    //Mostrando en pantalla la nueva matriz
    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            cout << matriz2[i][j] << " ";
        }
        cout << "\n";
    }

    system("pause");
    return 0;
}
```

**UNIVERSIDAD DE CIENCIAS EMPRESARIALES – UCEM PROGRAMACIÓN
ESTRUCTURADA EN C**

(IN-1100; 3 créditos)

Prof: Alberto Espinoza Zamora Cel: 8932-1139 a.esza23@gmail.com



**UNIVERSIDAD DE CIENCIAS EMPRESARIALES – UCEM PROGRAMACIÓN
ESTRUCTURADA EN C**

(IN-1100; 3 créditos)

Prof: Alberto Espinoza Zamora Cel: 8932-1139 a.esza23@gmail.com

