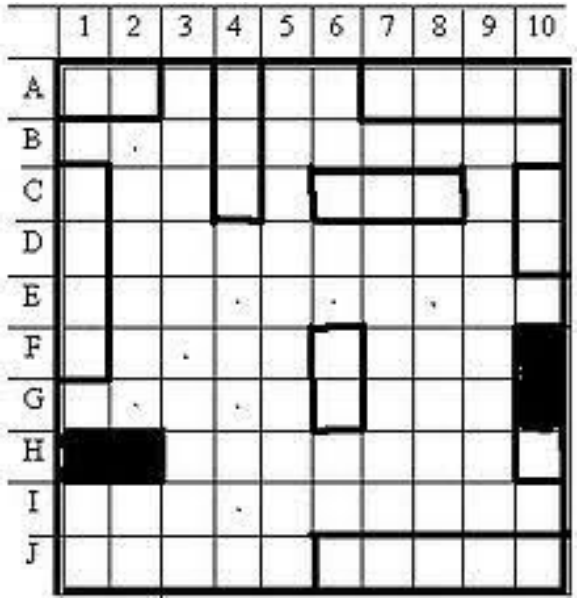
Array Multidimensional

Array Multidimensional

Una matriz multidimensional se puede denominar como una matriz de matrices que almacena datos homogéneos en forma tabular. Los datos en matrices multidimensionales se almacenan en orden de fila principal.



Array Multidimensional

La forma general de declarar arreglos N-dimensionales es:

tipo_datos nombre_arreglo[tamaño1][tamaño2]...[tamañoN];

data_type : tipo de datos que se almacenarán en la matriz.

array_name : Nombre de la matriz

size1, size2,… ,sizeN : Tamaños de la dimensión

Array Multidimensional

Ejemplos :

Matriz bidimensional: int two_d[10][20];

Matriz tridimensional: int three_d[10][20][30];

Array Multidimensional

El número total de elementos que se pueden almacenar en una matriz multidimensional se puede calcular multiplicando el tamaño de todas las dimensiones.

Por ejemplo:

La matriz int x[10][20] puede almacenar un total (10*20) = 200 elementos.

De manera similar, la matriz int x[5][10][20] puede almacenar un total (5*10*20) = 1000 elementos.

Matriz bidimensional

La matriz bidimensional es la forma más simple de una matriz multidimensional. Podemos ver una matriz bidimensional como una matriz de una matriz unidimensional para facilitar la comprensión.

La forma básica de declarar una matriz bidimensional de tamaño x, y:

Sintaxis:

tipo_datos nombre_arreglo[x][y];

Matriz bidimensional

Los elementos en matrices bidimensionales se denominan comúnmente x[i][j], donde i es el número de fila y 'j' es el número de columna.

Una matriz bidimensional se puede ver como una tabla con filas 'x' y columnas 'y' donde el número de fila varía de 0 a (x-1) y el número de columna varía de 0 a (y-1). A continuación se muestra una matriz bidimensional 'x' con 3 filas y 3 columnas:

Matriz bidimensional

	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]

Inicializar arreglos bidimensionales

Primer método :

int x[3][4] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

La matriz anterior tiene 3 filas y 4 columnas. Los elementos en las llaves de izquierda a derecha se almacenan en la tabla también de izquierda a derecha. Los elementos se completarán en el arreglo en orden, los primeros 4 elementos desde la izquierda en la primera fila, los siguientes 4 elementos en la segunda fila y así sucesivamente.

Inicializar arreglos bidimensionales

Segundo método :

```
int x[3][4] = {{0,1,2,3}, {4,5,6,7}, {8,9,10,11}};
```

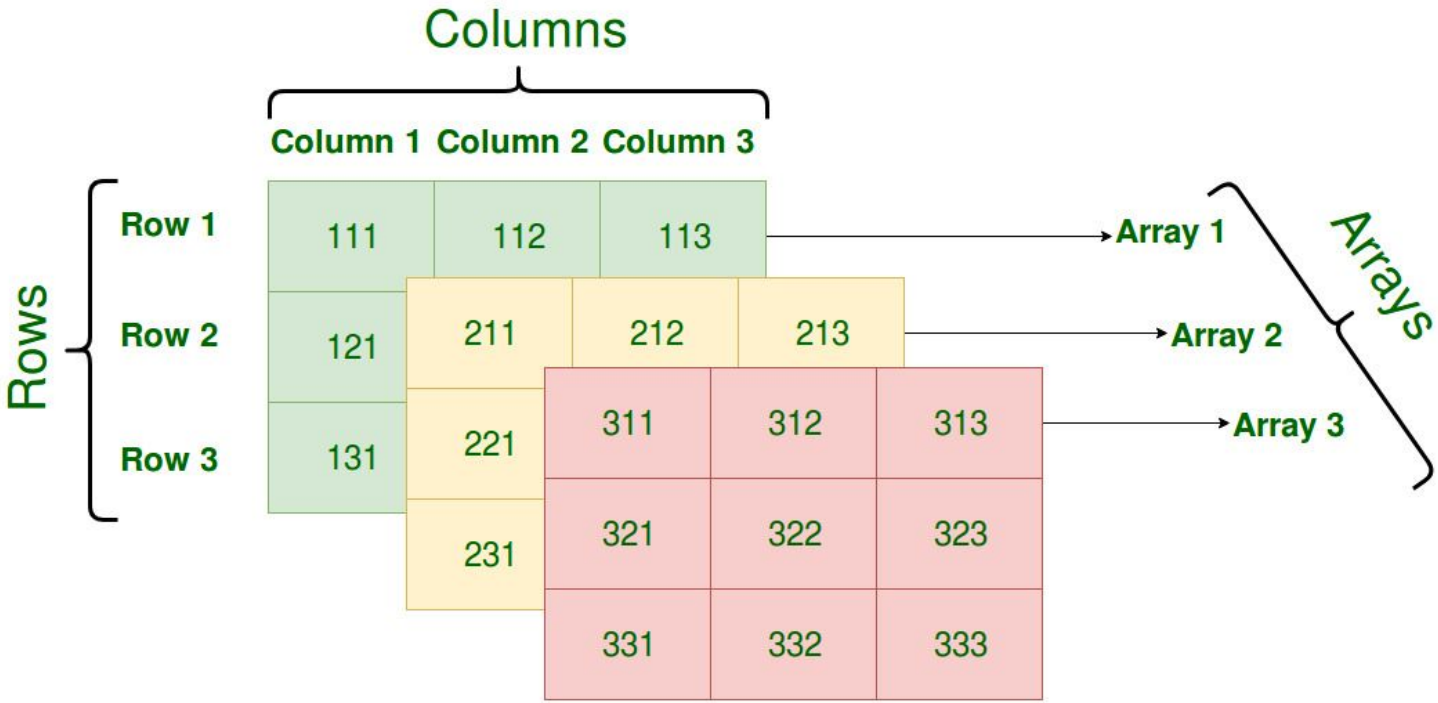
Tercer método:

```
int x[3][4];
for(int i = 0; i < 3; i++){
    for(int j = 0; j < 4; j++){
        cin >> x[i][j];
    }
}
```

Cuarto método Dinámico

```
int** x = new int*[3];
  for(int i = 0; i < 3; i++){
    x[i] = new int[4];
    for(int j = 0; j < 4; j++){
        cin >> x[i][j];
    }
}
   delete[]x;
```

Matriz tridimensional



Inicialización de una matriz tridimensional

La inicialización en una matriz tridimensional es la misma que la de las matrices bidimensionales. La diferencia es que a medida que aumenta el número de dimensiones, también aumentará el número de llaves anidadas.

Método 1 :

```
int x[2][3][4] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                  11, 12, 13, 14, 15, 16, 17, 18 , 19,
                  20, 21, 22, 23};
```

Método 2 (mejor) :

```
int  x[2][3][4] =
 {
   { {0,1,2,3}, {4,5,6,7}, {8,9,10,11} },
   { {12,13 ,14,15}, {16,17,18,19}, {20,21,22,23} }
 };
```

Números Aleatorios

Números Aleatorios

Una característica importante en los lenguajes de programación es la posibilidad de generar números aleatorios.

Los números aleatorios son pieza clave en la programación de juegos de azar, simulaciones, cifrado de datos, en programación de comportamiento pseudo inteligente, en modelos estadísticos, etc. Por esta razón es importante conocer las herramientas que un lenguaje de programación te ofrece en este sentido.

Función rand()

C++ define la función rand() para generar números aleatorios, está definida en la librería cstdlib.

rand() devuelve un número entero pseudo-aleatorio en el rango 0 y RAND_MAX. RAND_MAX está definida también en la librería cstdlib.

Sintaxis

```
int rand (void);
```

stdlib.h (std-lib: standard library o biblioteca estándar). Es el archivo de cabecera de la biblioteca estándar de propósito general del lenguaje de programación C. Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras. Es compatible con C++ donde se conoce como cstdlib.

Función rand()

De forma predeterminada rand() genera una valor pseudo aleatorio comprendido entre 0 y el valor de RAND_MAX, RAND_MAX está definido también el cstdlib.

Definiendo un límite superior para rand()

Para definir un límite superior, se utiliza la notación de módulo (%), a continuación se indica el valor máximo que se desea. por ejemplo.

```
valor = rand() % 2;          // Obtiene valores entre 0 y 1
```

Definiendo un límite superior para rand()

Por ejemplo, el código anterior permite simular el lanzamiento de una moneda, y en general cualquier cosa que requiera un si o un no.

```
valor = rand() % 10;     // Obtiene un número entre 0 y 9, nota que no incluye el 10valorrnd = 10 + rand() % 20;

valor = rand() % 1500;    // Obtiene un número entre 0 y 1499

valor = rand() % 65536 ;  // Obtiene un número entre 0 y 65535, por ejemplo,
        para elegir un puerto TCP/IP
```

Definiendo un límite inferior para rand()

Para definir el límite inicial utilizamos la notación de suma, indicando el número desde el cual se deben generar los números, por ejemplo:

```
valor = 10 + rand();        // Genera números aleatorios a partir del 10
valor = 65 + rand();        // Genera números aleatorios a partir del 65
```

Bibliografía

https://www.geeksforgeeks.org/multidimensional-arrays-c-cpp/

https://ehack.info/numeros-aleatorios-en-c/