



Concepto de Matriz

Es posible manejar arrays de más de una dimensión, denominados en general matrices. Cada dimensión está representada por un subíndice en la matriz. Por tanto, una matriz bidimensional tiene dos subíndices; una matriz tridimensional tiene tres subíndices; y así sucesivamente. Una matriz puede tener cualquier número de dimensiones, aunque las matrices más utilizadas son las de dos dimensiones.

Una matriz de dos dimensiones de tipo entera se define de la siguiente manera:

```
int Matriz [i] [j]
```

i: maneja las filas de la matriz

j: maneja las columnas de la matriz

Forma de inicializar una matriz es la siguiente:

```
tipo nombre_matriz[ Tamañofilas] [ TamañoColumnas] = {lista-valores};
```

Ejemplo:

```
int x[2] [3] = {3,5,7,-2,0,1};
```

Lenguaje de Programación C

La matriz anterior se puede inicializar de la siguiente manera:

```
int matriz1[5][3] = { 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 };
```

Para mayor claridad, se pueden agrupar los valores entre llaves. Por ejemplo,

```
int matriz1[5][3] = { {1,2,3},  
                      {4,5,6},  
                      {7,8,9},  
                      {10,11,12},  
                      {13,14,15} };
```

El compilador ignora las llaves interiores, que hacen más sencillo comprender cómo se distribuyen los valores. Cada valor debe separarse con una coma, independientemente de que se utilicen llaves o no. La inicialización completa si que debe encerrarse entre llaves, y debe finalizar con punto y coma.

Lenguaje de Programación C

```
1: #include <stdio.h>
2: int main()
3: {
4:     int i,j;
5:     int matriz[5][2] = { {0,0}, {1,2}, {2,4}, {3,6}, {4,8}};
6:
7:     for (i = 0; i<5; i++)
8:     {
9:         for (j=0; j<2; j++)
10:        {
11:            printf("matriz[%d][%d] = ", i, j);
12:            printf("%d \t", matriz[i][j]);
13:        }
14:        printf("\n");
15:    }
16:
17:    return 0;
18: }
```

Salida del programa

matriz[0][0] = 0	matriz[0][1] = 0
matriz[1][0] = 1	matriz[1][1] = 2
matriz[2][0] = 2	matriz[2][1] = 4
matriz[3][0] = 3	matriz[3][1] = 6
matriz[4][0] = 4	matriz[4][1] = 8

Realizar el programa para obtener una salida similar, pero para una matriz de 4x4

```
C:\Program Files (x86)\Zinja\bin\runner.exe
Ingresar un elemento
2
Ingresar un elemento
5
Ingresar un elemento
6
Ingresar un elemento
9
Ingresar un elemento
6
Ingresar un elemento
3
Ingresar un elemento
1
Ingresar un elemento
2
Ingresar un elemento
3
m[0][0]: 2
m[0][1]: 5
m[0][2]: 6
m[1][0]: 9
m[1][1]: 6
m[1][2]: 3
m[2][0]: 1
m[2][1]: 2
m[2][2]: 3

  2  5  6
  9  6  3
  1  2  3
```

A PROGRAMAR!!!!





**Calcular la sumatoria de los
elementos de la matriz**

A PROGRAMAR!!!!





**Calcular el promedio de los
elementos de la matriz**

A PROGRAMAR!!!!





**Ingresa un número por pantalla e
indicar cuantas veces aparece en la
matriz.**

A PROGRAMAR!!!!





Calcular la traspuesta de la matriz.

A PROGRAMAR!!!!





Agregar la función pares al programa anterior.

La función pares debe devolver al programa principal la cantidad de números pares

A PROGRAMAR!!!!

