

ARREGLOS MULTIDIMENSIONALES

Matrices

En matemática, una matriz es una disposición de números en filas y columnas:

$$\begin{pmatrix} 2 & 7 & -1 \\ 3 & 4 & 8 \\ -2 & 11 & 5 \end{pmatrix}$$

La matriz A tiene 3 filas y 3 columnas. Para denotar el elemento de la fila i columna j se escribe A_{ij} . Podemos escribir entonces en este ejemplo que $A_{12} = 7$

También al programar podemos necesitar de tablas de valores dispuestas en filas y columnas. Por ejemplo, para representar una tabla de posiciones, una tabla de tarifas u horarios de transportes, etc. Las matrices se usan entonces en programación y se conocen como arreglos multidimensionales.

Matrices en C

Un arreglo bidimensional de 3 filas y 3 columnas como el del ejemplo anterior se define en C indicando el tipo, el nombre del arreglo y entre corchetes la cantidad de filas y la cantidad de columnas:

```
int array [3][3];
```

Se puede inicializar en el mismo momento, de la siguiente manera:

```
int array [3][3] = {{3, 4, 5},  
                    {5, 7, 8},  
                    {4, 3, 5}};
```

Para hacer referencia a un elemento particular, entre corchetes se indica la fila y la columna:

```
array[0][0] vale 3
```

```
array[0][1] vale 4
```

MUY IMPORTANTE: La indización **comienza en 0** y es igual que en los vectores.

Funciones: Generar y mostrar una matriz en C

Detallamos las funciones necesarias para generar una matriz con datos ingresados por teclado y mostrarla por pantalla.

```
void GenerarMatriz (int A[][8], int filas, int cols){
```

```
    int i, j;
```

```
    for (i = 0; i < filas; i++){
```

```
    {
```

```
        for (j = 0; j < cols; j++){
```

```
        {
```

```
            scanf("%d",&num);
```

```
            A[i][j] = num;
```

```
        }
```

```
    }
```

```
}
```

Diferencia con un arreglo: En la matriz, al pasarla por parámetro, debe tener el número de columnas, sino es un error de compilación

```
void MostrarMatriz (int A[][8], int filas, int cols){
```

```
    int i, j;
```

```
    for (i = 0; i < filas; i++){
```

```
    { printf("\n");
```

```
        for (j = 0; j < cols; j++){
```

```
        {
```

```
            printf("%d\t",A[i][j]);
```

```
        }
```

```
    }
```

```
}
```

Para que cada fila se vea es un renglón diferente

\t de tabulación, deja un espacio entre los números de la matriz

Programa principal

```

int main()

{ int A[10][10], filas, cols;

  printf("ingrese cantidad de filas");

  scanf("%d",&filas);

  printf("ingrese cantidad de columnas");

  scanf("%d",&cols);

  GenerarMatriz (A, filas, cols);

  MostrarMatriz (A, filas,cols)

}

```

TIPOS DE MATRICES

Traspuesta

Dada una matriz A, se llama traspuesta de A a la matriz que se obtiene cambiando ordenadamente las filas por las columnas.

Se representa por A^t ó A^T

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad A^t = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

Opuesta

La matriz opuesta de una dada es la que resulta de sustituir cada elemento por su opuesto. La opuesta de A es -A.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad -A = \begin{pmatrix} -1 & -2 \\ -3 & -4 \end{pmatrix}$$

Cuadrada

Aquella matriz que tiene igual número de filas que de columnas.

Diagonal principal:

$$A = \begin{pmatrix} 5 & 1 & 5 & -6 \\ 1 & 2 & 0 & 3 \\ 7 & -3 & 4 & 11 \\ 1 & 9 & 3 & 8 \end{pmatrix}$$

Diagonal secundaria:

$$A = \begin{pmatrix} 5 & 1 & 5 & -6 \\ 1 & 2 & 0 & 3 \\ 7 & -3 & 4 & 11 \\ 1 & 9 & 3 & 8 \end{pmatrix}$$

Simétrica

Es una matriz cuadrada que es igual a su traspuesta.

Ejemplo:

$$A_3 = \begin{pmatrix} 1 & 9 & -6 \\ 9 & 2 & 1 \\ -6 & 1 & 5 \end{pmatrix}$$