

TC1017

Solución de problemas con programación

Matrices

ITESM Campus Querétaro



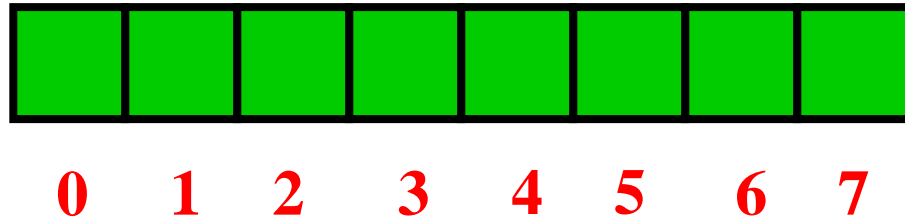
La sesión pasada vimos...



- Arreglos

Matriz

Un **arreglo** lo definimos como **la relación entre un nombre y un conjunto de localidades**. Decimos que la estructura de datos que definimos en el arreglo es de una sola dimensión ya que utilizamos un solo valor para identificar a cada localidad (0,1,2,...,n-1).



Matriz

Una **matriz** también es una colección de localidades asociadas a un nombre, sólo que los datos se organizan en **dos dimensiones**. Por ello, para hacer referencia a una localidad del arreglo se necesitan de dos números:

- * El **número de renglón**
- * El **número de columna**

	0	1	2	3	4	5
0						
1						
2						
3						

Matriz

- De la misma manera que en los arreglos, la numeración de renglones y de columnas inicia desde **0**.
- Si por ejemplo definimos una matriz de **6** renglones y **6** columnas, el primer elemento de la colección se encontraría en:
 - **renglón 0**
 - **columna 0**
- Y el último elemento se encontraría en:
 - **renglón 5**
 - **columna 5**



Definición de una matriz

- La definición de variables de tipo matriz es similar a los arreglos, la forma general de declarar una variable matriz es la siguiente:

tipo **nombre** **[#renglones][#columnas];**

1	2	3
4	5	6
7	8	9

- tipo** es un tipo de datos (int, double, char).
- nombre** es el nombre de la variable.
- #renglones** es el número de renglones de la matriz.
- #columnas** el número de columnas de la matriz.

Localidades de una matriz

- Una vez que hemos declarado la variable matriz, **¿Cómo tenemos acceso a los valores?** indicando el **renglón** y la **columna** de dicha localidad.
- Para esto debemos recordar que los **renglones** están numerados de **0** a **r-1** (en donde r es el número de renglones del arreglo) y que las **columnas** están numeradas de **0** a **c-1** (en donde c es el número de columnas del arreglo).

1	2	3
4	5	6
7	8	9

Localidades de una matriz

- Para hacer referencia a una localidad específica del arreglo debemos escribir el nombre de la variable y entre corchetes el número de renglón y el número de columna de la localidad.

`int M[8][6];`

- Así por ejemplo, la manera como referenciamos a la primera localidad del arreglo M que se definió anteriormente es `M[0][0]` y la última localidad es `M[7][5]`

1	2	3
4	5	6
7	8	9

Asignaciones a una localidad de la matriz

- La forma de asignar un valor a una localidad específica de la matriz es la siguiente:

nombre[**renglon**][**columna**] = **valor**;

- en donde **nombre** es el nombre de la variable matriz, **renglon** es el número del renglón de la localidad y **columna** el número de columna de la localidad de la matriz y **valor** es cualquier valor del tipo con que fue definida la matriz.

1	2	3
4	5	6
7	8	9

Actividad Grupal

Realizar ejercicios con matrices para comprender su funcionamiento.



¿Cómo quedaría la matriz siguiente al ejecutar las instrucciones descritas?

```
int M[4][4];
```

```
int x1=10;
```

```
int x2=20;
```

```
M[0][0] = X1;
```

```
M[1][0] = X2;
```

```
M[0][3] = M[0][0] * x2;
```

	0	1	2	3
0	10			?
1	20			
2				
3				



¿Cual sería el valor de X dada la siguiente matriz?

```
int M[4][4];
```

```
X= M[0][0]+M[0][1]+ M[0][2]+M[0][3]
```

	0	1	2	3
0	10	20	30	40
1	20	10	20	20
2	30	20	10	20
3	40	20	20	10



Actividad colaborativa

(3 minutos)

Escribe las instrucciones necesarias para poder llenar la diagonal de una matriz con ceros, la matriz es de dimensión 4 X 4



	0	1	2	3
0	0			0
1		0	0	
2		0	0	
3	0			0

Matriz

La estructura compañera de las matrices es **dos ciclos for anidados**. Como te darás cuenta, la mayoría de los problemas de matrices tendrán un código similar al siguiente:

```
int i, j;  
for (i=0; i<renglones; i++)  
{  
    for (j=0; j<columnas; j++)  
    {  
  
    }  
}
```



For anidados

- Esta estructura de doble ciclo nos permite recorrer todas las localidades el arreglo, con el primer ciclo **for** se recorren los **renglones**, y con el segundo ciclo se recorren las **columnas**.
- En el primer ciclo se recorre cada **renglón**, para ello la variable del ciclo (**i**) toma los valores:

0, 1, 2, ... , renglones – 1

que son precisamente los números de cada renglón de la matriz.

For anidados

- En el segundo ciclo para el renglón **i**, se recorre cada **columna** de la matriz, para ello la variable del ciclo (**j**) toma los valores:

0, 1, 2, ... , columnas-1

que son precisamente los números de cada columna de la matriz.

1	2	3
4	5	6
7	8	9

Actividad Grupal

Escriba el código del procedimiento **iniciaMatriz**, que recibe una matriz de enteros de **3** renglones y **3** columnas y le asigna a cada localidad el valor de **5**.



```
#define renglones 3
```

```
#define columnas 3
```

```
void iniciaMatriz (int M[renglones][columnas])
```

```
{
```

```
    int i, j;
```

```
    for (i=0; i<renglones; i++)
```

```
    {
```

```
        for (j=0; j<columnas; j++)
```

```
        {
```

```
            M[ i ][ j ] = 5;
```

```
        }
```

```
    }
```

```
}
```



Solución

Actividad Grupal

Escriba el código del procedimiento **imprimeMatriz**, que recibe una matriz de enteros de **3** renglones y **3** columnas y despliega en pantalla el contenido de la matriz.



```
#define renglones 3
```

```
#define columnas 3
```

```
void imprimeMatriz (int M[renglones][columnas])  
{  
    int i, j;  
    for (i=0; i<renglones; i++)  
    {  
        for (j=0; j<columnas; j++)  
        {  
            printf("%i ", M[ i ][ j ]);  
        }  
        printf("\n");  
    }  
}
```



Solución

Actividad colaborativa

(3 minutos)

Escriba el código del procedimiento **iniciaMatriz2**, que recibe una matriz de enteros de **3** renglones y **3** columnas y le asigna a cada localidad un número consecutivo correspondiente del **1** al **9**



	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

```
#define renglones 3
```

```
#define columnas 3
```

```
void iniciaMatriz2 (int M[renglones][columnas])  
{  
    int aux=1, i, j;  
    for (i=0; i<renglones; i++)  
    {  
        for (j=0; j<columnas; j++)  
        {  
            M[ i ][ j ] = aux;  
            aux++;  
        }  
    }  
}
```



Solución

Actividad colaborativa

(10 minutos)



- Escriba el código del procedimiento **sumaMatrices**, que recibe las matrices A, B y C de enteros de **3** renglones y **3** columnas cada una.
- El procedimiento asignará en la localidad correspondiente de la matriz C la suma de las matrices A más B.

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 9 & 10 & 11 \end{bmatrix} + \begin{bmatrix} 1 & 3 & 5 \\ 9 & 11 & 13 \\ 17 & 19 & 21 \end{bmatrix} = \begin{bmatrix} 2 & 5 & 8 \\ 14 & 17 & 20 \\ 26 & 29 & 32 \end{bmatrix}$$

```
#define renglones 3
```

```
#define columnas 3
```

```
void sumaMatrices (int A[renglones][columnas],  
                   int B[renglones][columnas],  
                   int C[renglones][columnas])
```

```
{  int i, j;  
    for (i=0; i<renglones; i++)  
    {  
        for (j=0; j<columnas; j++)  
        {  
            C[ i ][ j ]= A[ i ][ j ] + B[ i ][ j ];  
        }  
    }  
}
```



Solución

Actividad Grupal

Integrar todos los procedimientos vistos anteriormente en un solo programa para verificar su funcionamiento



¿Cómo crear una matriz?

Con el método insert

```
matriz = []  
for i in range(0, renglones):  
    matriz.insert(i, [ ])  
    for j in range(0, columnas):  
        valor = int(input("Introduce un valor: "))  
        matriz[i].insert(j, valor)
```

Con el método append

```
matriz = []  
for i in range(0, renglones):  
    matriz.append([ ])  
    for j in range(0, columnas):  
        valor = int(input("Introduce un valor: "))  
        matriz[i].append(valor)
```

