



Programación 1

Problemas con matrices

En los siguientes problemas se va a trabajar con matrices de enteros, que van a ser de una dimensión máxima de $DIM \times DIM$, siendo DIM una constante simbólica establecida en el código. Sin embargo, el usuario establecerá, cuando ejecute el programa, la dimensión exacta de las matrices (que tendrán que estar entre 0 y DIM , tanto para el número de filas como para el número de columnas). De esta forma, se trabajará con unas matrices *sobredimensionadas*: el número de filas y columnas de la matriz declarada puede ser mayor que el número de filas y columnas con las que se vaya a trabajar.

Por ejemplo, con $DIM = 10$, el usuario podría querer trabajar con una matriz de dimensiones 3×4 . Una representación gráfica de dicha matriz podría ser la siguiente:

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

1. Leer una matriz del teclado

Diseña una función `leerMatriz` que:

- solicite al usuario la dimensión de la matriz, donde nf es el número de filas y nc , el de columnas,
- verifique que ambos números sean positivos y menores o iguales a DIM . Mientras no se cumplan estas condiciones, vuelve a pedir los datos:
- Una vez que nf y nc cumplen con las condiciones, solicita al usuario los $nf \times nc$ datos enteros que sirven para inicializar las primeras $nf \times nc$ componentes de la matriz (de menor a mayor índice).

```
/*  
 * Pre: La matriz «mat» tiene unas dimensiones máximas de  $DIM \times DIM$ .  
 * Post: Se ha solicitado al usuario y el número de filas «nf» y de columnas «nc»  
 *       y se ha leído del teclado hasta que  $0 < nf \leq DIM$  y  $0 < nc \leq DIM$ . A  
 *       continuación, se ha solicitado la introducción por el teclado de  
 *        $nf \times nc$  datos enteros y se han copiado en las primeras  $nf \times nc$   
 *       componentes de «mat», de menor a mayor índice.  
 */  
void leerMatriz(int mat[][DIM], unsigned int& nf, unsigned int& nc);
```



2. Escribir una matriz en la pantalla

Diseña una función denominada `escribirMatriz` que escriba en la pantalla las primeras $nf \times nc$ componentes de una matriz `mat` de dimensión máxima $DIM \times DIM$. **Completa antes su cabecera:**

```
/*
 * Pre: La matriz «mat» de dimensiones máximas DIM x DIM tiene las primeras
 *       «nf» x «nc» componentes inicializadas, con  $0 < nf \leq DIM$  y  $0 < nc \leq DIM$ .
 * Post: Ha escrito los valores de las «nf» x «nc» primeras componentes de la
 *       matriz «mat» desde la de menor índice a la de mayor.
 */
_____ escribirMatriz(_____ mat_____, _____ nf, _____ nc);
```

3. Traspuesta de una matriz

Completa la cabecera y diseña la siguiente función:

```
/*
 * Pre: La matriz «mat» es una matriz de dimensiones máximas DIM x DIM
 *       y «nf» y «nc» están comprendidos entre 0 y DIM (inclusive).
 * Post: Ha escrito la traspuesta de la matriz «mat» en la pantalla.
 */
_____ escribirTraspuesta(_____ mat_____, _____ nf, _____ nc);
```

4. Simetría de una matriz

Escribe la especificación, cabecera y cuerpo de una función denominada `esSimetrica` que, dadas una matriz M de dimensiones máximas $DIM \times DIM$ y un entero n comprendido entre 1 y DIM que indica la dimensión exacta de dicha matriz, indique si la matriz M de dimensiones $n \times n$ es simétrica o no lo es.

5. Multiplicación de dos matrices

Diseña la siguiente función:

```
/*
 * Pre: Las matrices «A», «B» y «C» son cuadradas de DIM x DIM y «nfa», «nca», «nfb»
 *       y «ncb» están comprendidos entre 0 y DIM (inclusive).
 *       Se cumplen las condiciones para la multiplicación de las dos matrices
 *       «A» y «B»:  $nca = nfb$ .
 * Post: La matriz «C» es la matriz producto de la multiplicación  $A \times B$ .
 */
void multiplicar (const int A[][DIM], const int B[][DIM], const unsigned int nfa,
                  const unsigned int nca, const unsigned int ncb, int C[][DIM]);
```



6. Programa principal

Diseña un programa C++ para verificar el correcto funcionamiento de las funciones previas. En particular, el programa:

1. Pide al usuario introducir por el teclado una matriz M de dimensión $n \times m$ (la dimensión máxima permitida es 10×10) y escribe su matriz traspuesta en la pantalla.
2. Si la matriz M es cuadrada, verifica si M es simétrica y escribe en la pantalla el mensaje correspondiente.
3. Pide al usuario introducir por el teclado dos matrices A (de dimensión $nfa \times nca$) y B (dimensión $nfb \times ncb$).
4. Controla si se puede calcular el producto de A y B (es decir, si $nca = nfb$). Si se puede, calcula la matriz producto C (de dimensión $nfa \times ncb$) y escribe C en la pantalla. En otro caso, avisa al usuario con un mensaje.

Ejemplo de ejecución:

TRASPUESTA DE UNA MATRIZ:

Introduzca el número de filas y columnas (>0): 3 4

Fila 1: Introduzca 4 enteros: 2 0 1 9

Fila 2: Introduzca 4 enteros: 0 8 6 7

Fila 3: Introduzca 4 enteros: 1 6 9 3

La matriz traspuesta es:

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

MULTIPLICACIÓN DE DOS MATRICES:

Introduzca el número de filas y columnas (>0): 4 2

Fila 1: Introduzca 2 enteros: 1 2

Fila 2: Introduzca 2 enteros: 3 4

Fila 3: Introduzca 2 enteros: 5 6

Fila 4: Introduzca 2 enteros: 7 8

Introduzca el número de filas y columnas (>0): 2 6

Fila 1: Introduzca 6 enteros: -1 2 3 4 5 6

Fila 2: Introduzca 6 enteros: 1 -2 3 4 5 6

El producto es:

1	-2	9	12	15	18
1	-2	21	28	35	42
1	-2	33	44	55	66
1	-2	45	60	75	90