

PROGRAMACIÓN I - GRUPO A

PRÁCTICA Nº5

1 de diciembre 2022

INSTRUCCIONES

1. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de Programación I antes de la hora de finalización de la sesión de prácticas.
2. Aunque las prácticas se pueden realizar en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual**. En otro caso, la práctica quedará **sin evaluar** y supondrá un **0 en su calificación**.
3. Se debe entregar **un único fichero** en **formato .txt o .cpp, sin comprimir**, con el nombre de **PracticaA5**.
4. El fichero entregado debe **incluir el nombre de los integrantes del equipo**.
5. Durante el desarrollo de la sesión práctica se debe seguir el procedimiento de trabajo en la sala asignada para realizar las actividades evaluables. Recordad que es **imprescindible GRABAR** toda la sesión y, además, **COMPARTIR el escritorio y ACTIVAR la cámara web y el micrófono**.
6. Se debe acceder a la **SALA DE PRÁCTICA**, desde donde se darán indicaciones y se resolverán dudas generales. Si lo consideráis necesario, utilizad **el chat** de esta sala para **comunicaros con el profesor/a**, indicando vuestro **número de grupo o nombre/apellidos**.
7. El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su **descalificación**.

Escriba un programa en C++11, **PracticaA5**, utilizando los **tipos de datos**, las **estructuras de control** y las **funciones** necesarias, que muestre por pantalla un **menú** con las siguientes opciones:

1. **Elementos maximo y minimo de una matriz.**
2. **Crear vector filas de la matriz.**
3. **Crear vector columnas de la matriz.**
4. **Multiplicar dos matrices.**
5. **Crear la traspuesta de una matriz.**
6. **Salir programa.**

El menú será una **función** (llamada **menu**) que, **sin recibir ningún parámetro**, devolverá la **opción seleccionada por el usuario**. La función solicitará la opción al usuario que tendrá que validarse. Así, si la **opción indicada por el usuario no es válida**, se mostrará el mensaje “**La opción no es válida. Por favor, vuelva a seleccionar otra opción**” y **presentará de nuevo el menú anterior** para que el usuario pueda **repetir el proceso** seleccionando otra opción hasta que sea válida **(0,25 puntos)**.

Siempre que se seleccione cualquiera de las **cinco primeras opciones** el programa, desde la función **main**, deberá realizar y ejecutar dicha operación. Después, volverá a **presentar de nuevo el menú anterior**, llamando a la función **menu**, para que el usuario pueda **repetir el proceso** seleccionando otra opción. Si el usuario selecciona la **opción 6**, el programa terminará mostrando el mensaje: “**Gracias y hasta pronto**” **(0,25 puntos)**.

Importante: Las diferentes opciones deberán ser resueltas de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general y no particular** para unos **determinados datos**.

La descripción de las opciones del menú es la siguiente:

1. Elementos maximo y minimo de una matriz. (3 puntos)

Esta opción se tendrá que realizar a través de **funciones**. Su objetivo es determinar los **valores máximo y mínimo** de todos los almacenados en una **matriz solicitada al usuario**. Finalmente, desde el programa principal (función **main**) se mostrará un **mensaje para indicar los valores determinados, el máximo y el mínimo**.

Las tareas que hay que realizar son las siguientes:

- Diseñar e implementar la función de nombre **solicitarDimension**. Esta función devolverá la dimensión de la matriz que se solicitará al usuario. La dimensión tendrá que ser un valor entero positivo comprendido entre 2 y 100 **(0,5 puntos)**.
- Diseñar e implementar la función de nombre **solicitarMatriz**. Esta función recibirá la dimensión indicada por el usuario anteriormente y devolverá una matriz cuadrada de dicha dimensión (**Matriz**) y de tipo **vector**. La función solicitará al usuario los datos que se guardarán en una matriz cuadrada que será una variable de tipo **vector (Matriz)** **(1 punto)**.
- Diseñar e implementar la función de nombre **mostrarMatriz**. Esta función recibirá como parámetro la matriz solicitada al usuario anteriormente y mostrará su contenido en formato de matriz (filas x columnas, ver ejemplo de ejecución) **(0,5 puntos)**.
- Diseñar e implementar la función de nombre **obtenerMax_Min**. Esta función recibirá como parámetro la matriz solicitada al usuario anteriormente para determinar los **valores máximo y mínimo** que hay en ella y que serán devueltos por la función para que desde el **main** se pueda mostrar el mensaje **(1 punto)**:

El valor Maximo es MM y el valor Minimo es mm.

Los valores de la matriz pueden estar repetidos, pero solo importa saber cuál es el mayor y el menor de los existentes en ella.

Ejemplo de ejecución para una matriz de 3x3:

```

Por favor, indique la dimension de la matriz (entre 2 y 100): 3

DATOS DE LA MATRIZ
=====
Elemento[1, 1] 1
Elemento[1, 2] -2
Elemento[1, 3] 3
Elemento[2, 1] -4
Elemento[2, 2] 5
Elemento[2, 3] -6
Elemento[3, 1] 7
Elemento[3, 2] -8
Elemento[3, 3] 9

MATRIZ SOLICITADA AL USUARIO
| 1 -2 3 |
| -4 5 -6 |
| 7 -8 9 |

El valor Maximo es 9 y el valor Minimo es -8

```

2. Crear vector filas de la matriz. (1,5 puntos)

Esta opción se tendrá que realizar a través de **funciones**. Su objetivo es crear una matriz unidimensional para almacenar la suma de los elementos de cada fila de una matriz cuadrada solicitada al usuario.

Las tareas que hay que realizar son las siguientes:

- Solicitar la dimensión utilizando la función de nombre **solicitarDimension**.
- Solicitar al usuario los datos de una matriz (de tipo **vector**) utilizando la función de nombre **solicitarMatriz**.
- Mostrar el contenido de la matriz en formato de matriz (filas x columnas, ver ejemplo de ejecución) utilizando la función de nombre **mostrarMatriz**.
- Diseñar e implementar la función de nombre **obtenerVectorFilas**. Esta función recibirá como parámetro la matriz solicitada al usuario anteriormente y devolverá una nueva matriz (**mFilas**) de tipo vector y con el mismo número de filas que la matriz recibida. La función calculará los valores de la matriz **mFilas**, donde cada elemento es el resultado de **sumar los valores ubicados en cada fila** de matriz recibida como parámetro (1 punto).
- Diseñar e implementar la función de nombre **mostrarVector**. Esta función recibirá como parámetro la matriz calculada en el apartado anterior y la mostrará su contenido por pantalla en formato matriz (ver ejemplo de ejecución) (0,5 puntos).

Ejemplo de ejecución para una matriz de 3x3:

```
Por favor, indique la dimension de la matriz (entre 2 y 100): 3
```

```
DATOS DE LA MATRIZ
=====
Elemento[1, 1] 1
Elemento[1, 2] 2
Elemento[1, 3] 3
Elemento[2, 1] 4
Elemento[2, 2] 5
Elemento[2, 3] 6
Elemento[3, 1] 7
Elemento[3, 2] 8
Elemento[3, 3] 9
```

```
MATRIZ SOLICITADA AL USUARIO
```

```
| 1 2 3 |
| 4 5 6 |
| 7 8 9 |
```

```
MATRIZ SUMA FILAS
```

```
| 6 |
| 15 |
| 24 |
```

3. Crear vector columnas de la matriz. (1 punto)

Esta opción se tendrá que realizar a través de **funciones**. Su objetivo es crear una matriz unidimensional para almacenar la suma de los elementos de cada columna de una matriz cuadrada solicitada al usuario.

Las tareas que hay que realizar son las siguientes:

- Solicitar la dimensión utilizando la función de nombre **solicitarDimension**.
- Solicitar al usuario los datos de una matriz (de tipo **vector**) utilizando la función de nombre **solicitarMatriz**.
- Mostrar el contenido de la matriz anterior en formato de matriz (filas x columnas, ver ejemplo de ejecución) utilizando la función de nombre **mostrarMatriz**.
- Diseñar e implementar la función de nombre **obtenerVectorColumnas**. Esta función recibirá como parámetro la matriz solicitada al usuario anteriormente y devolverá una nueva matriz (**mColumnas**) de tipo vector y con el mismo número de columnas que la matriz recibida. La función calculará los valores de la matriz **mColumnas**, donde cada elemento es el resultado de **sumar los valores ubicados en cada columna** de matriz recibida como parámetro (1 punto).
- Mostrar el contenido de la nueva matriz creada en el apartado anterior utilizando la función de nombre **mostrarVector**.

Ejemplo de ejecución para una matriz de 3x3:

```
Por favor, indique la dimension de la matriz (entre 2 y 100): 3
```

```
DATOS DE LA MATRIZ
=====
Elemento[1, 1] 1
Elemento[1, 2] 2
Elemento[1, 3] 3
Elemento[2, 1] 4
Elemento[2, 2] 5
Elemento[2, 3] 6
Elemento[3, 1] 7
Elemento[3, 2] 8
Elemento[3, 3] 9
```

```
MATRIZ SOLICITADA AL USUARIO
| 1 2 3 |
| 4 5 6 |
| 7 8 9 |
```

```
VECTOR SUMA COLUMNAS
| 12 |
| 15 |
| 18 |
```

4. Multiplicar dos matrices. (2 puntos)

Esta opción se tendrá que realizar a través de **funciones**. Su objetivo es crear una matriz bidimensional para almacenar el producto de dos matrices cuadradas que se solicitan al usuario.

Las tareas que hay que realizar son las siguientes:

- Solicitar la dimensión utilizando la función de nombre **solicitarDimension**.
- Solicitar al usuario los datos de la **primera matriz** (de tipo **vector**) utilizando la función de nombre **solicitarMatriz**.
- Solicitar al usuario los datos de la **segunda matriz** (de tipo **vector**) utilizando la función de nombre **solicitarMatriz**.
- Mostrar el contenido de las matrices anteriores en formato de matriz (filas x columnas, ver ejemplo de ejecución) utilizando la función de nombre **mostrarMatriz**.
- Diseñar e implementar la función de nombre **obtenerProducto**. Esta función recibirá como parámetros las dos matrices solicitadas al usuario anteriormente y devolverá una nueva matriz (**mProducto**) de tipo vector y con igual dimensión que las matrices recibidas. La función calculará los valores de la nueva matriz, **mProducto**, donde cuyos elementos se obtendrán de realizar la multiplicación de las matrices recibidas como parámetro (2 puntos).
- Mostrar el contenido de la nueva matriz, matriz producto, en formato de matriz (filas x columnas, ver ejemplo de ejecución) utilizando la función de nombre **mostrarMatriz**.

Ejemplo de ejecución para dos matrices de 3x3:

```

Por favor, indique la dimension de la matriz (entre 2 y 100): 3

DATOS DE LA MATRIZ
=====
Elemento[1, 1] 1
Elemento[1, 2] 2
Elemento[1, 3] 3
Elemento[2, 1] 4
Elemento[2, 2] 5
Elemento[2, 3] 6
Elemento[3, 1] 7
Elemento[3, 2] 8
Elemento[3, 3] 9

DATOS DE LA MATRIZ
=====
Elemento[1, 1] 1
Elemento[1, 2] -1
Elemento[1, 3] 1
Elemento[2, 1] -1
Elemento[2, 2] 1
Elemento[2, 3] -1
Elemento[3, 1] 1
Elemento[3, 2] -1
Elemento[3, 3] 1

PRIMERA MATRIZ SOLICITADA AL USUARIO
| 1 2 3 |
| 4 5 6 |
| 7 8 9 |

SEGUNDA MATRIZ SOLICITADA AL USUARIO
| 1 -1 1 |
| -1 1 -1 |
| 1 -1 1 |

MATRIZ PRODUCTO
| 2 -2 2 |
| 5 -5 5 |
| 8 -8 8 |

```

5. Crear la traspuesta de una matriz. (2 puntos)

Esta opción se tendrá que realizar a través de **funciones**. Su objetivo es crear una matriz bidimensional para almacenar la matriz traspuesta de otra matriz cuadrada que se solicita al usuario.

Las tareas que hay que realizar son las siguientes:

- Solicitar la dimensión utilizando la función de nombre **solicitarDimension**.
- Solicitar al usuario los datos de una matriz (de tipo **vector**) utilizando la función de nombre **solicitarMatriz**.
- Mostrar el contenido de la matriz anterior en formato de matriz (filas x columnas, ver ejemplo de ejecución) utilizando la función de nombre **mostrarMatriz**.
- Diseñar e implementar la función de nombre **obtenerTraspuesta**. Esta función recibirá como parámetro la matriz solicitada al usuario anteriormente y devolverá una nueva matriz (**mTraspuesta**) de tipo vector y con igual dimensión que la matriz recibida. La función obtendrá la nueva matriz (**mTraspuesta**) donde los elementos son los mismos que los de la matriz recibida como parámetro, pero cambiando filas por columnas (2 puntos).
- Mostrar el contenido de la matriz traspuesta en formato de matriz (filas x columnas, ver ejemplo de ejecución) utilizando la función de nombre **mostrarMatriz**.

Ejemplo de ejecución para dos matrices de 3x3:

```
Por favor, indique la dimension de la matriz (entre 2 y 100): 3
```

```
DATOS DE LA MATRIZ
=====
Elemento[1, 1] 1
Elemento[1, 2] 2
Elemento[1, 3] 3
Elemento[2, 1] 4
Elemento[2, 2] 5
Elemento[2, 3] 6
Elemento[3, 1] 7
Elemento[3, 2] 8
Elemento[3, 3] 9
```

```
MATRIZ SOLICITADA AL USUARIO
```

```
| 1 2 3 |
| 4 5 6 |
| 7 8 9 |
```

```
MATRIZ TRASPUESTA
```

```
| 1 4 7 |
| 2 5 8 |
| 3 6 9 |
```