

ALUMNO:

EPS -Ingeniería Informática

Asignatura: G0460006 Programación I

Curso: 2022/2023
Semestre: 1º

Examen: Final
Convocatoria: Ordinaria

Fecha: 16-1-2023

Parte Práctica (10 puntos; 70% nota final) - Tiempo: 2h40m

Los ficheros correspondientes a los ejercicios se deben entregar en la actividad correspondiente a través del campus antes de la finalización del tiempo establecido. Cada fichero se llamará EjercicioX, donde la X será el número de dicho ejercicio. Los únicos formatos válidos serán **txt** o **cpp**, siempre **sin comprimir**.

Criterios generales de evaluación

Funciones/Métodos: Si no se usa el paso por referencia constante cuando las variables de los parámetros de entrada no son de tipo simple.	40%
Tipos de datos y variables: <ul style="list-style-type: none">• Uso de variables globales (fuera del ámbito de una función).• Si no se usan los tipos contenedor vistos en clase (<code>std::array</code>; <code>std::vector</code>; <code>std::set</code>; <code>std::string</code>, etc.) para las variables que lo necesiten.	0%
El programa no compila o no se asemeja a lo pedido.	0%
Si no se cumplen los criterios de entrega indicados en la actividad/examen .	0%

Criterios particulares de evaluación

El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegidos son adecuados.	100%

IMPORTANTE:

- Todos los ejercicios del examen deberán ser resueltos de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general** y **no particular** para unos determinados datos/valores.
- Todos los ejercicios resueltos sin utilizar funciones cuando sea apropiado se valorarán con una nota máxima del 60% sobre la calificación prevista.

- Se recomienda una primera lectura del examen completo para planificar la realización del examen. Y una segunda lectura detallada antes de la realización de cada uno de los ejercicios propuestos.

Ejercicio 1 (3 puntos)

Escriba un programa en C++11, *Ejercicio1*, utilizando los tipos de datos, las **estructuras de control** y las **funciones** necesarias para determinar si una matriz solicitada al usuario es *triangular* y qué *tipo* es, *triangular superior* o *triangular inferior*.

Una *matriz triangular* es una **matriz cuadrada** en la que todos los elementos por encima o por debajo de la diagonal principal son *iguales a cero*. La *matriz triangular superior* es una matriz cuadrada cuyos elementos por debajo de la diagonal principal son *cero*. La *matriz triangular inferior* es una matriz cuadrada cuyos elementos por encima de la diagonal principal son *cero*.

El proceso debe realizar las siguientes tareas:

- Solicitar al usuario números enteros para crear una matriz cuadrada de dimensión definida a través del **valor de una constante declarada** en la función *main* del programa (**0,75 puntos**).
- Determinar si dicha matriz es triangular. En caso afirmativo, determinar si es triangular superior o inferior (**1 punto**).
- Mostrar por pantalla el contenido de la matriz cuadrada indicando si es una matriz triangular, superior o inferior, o no es triangular (**0,75 puntos**).
- Finalmente, se preguntará al usuario si desea repetir el proceso con una nueva matriz o finalizar la ejecución (**0,5 puntos**).

A continuación, se muestran algunos ejemplos de ejecución:



DATOS DE LA MATRIZ

Elemento[1, 1] 1

Elemento[1, 2] 0

Elemento[1, 3] 0

Elemento[2, 1] 2

Elemento[2, 2] 1

Elemento[2, 3] 3

Elemento[3, 1] 4

Elemento[3, 2] 5

Elemento[3, 3] 1

La matriz

	1	0	0	
	2	1	3	
	4	5	1	

NO es triangular.

Desea repetir el proceso con otros datos (S/N)? s

DATOS DE LA MATRIZ

Elemento[1, 1] 1

Elemento[1, 2] 0

Elemento[1, 3] 0

Elemento[2, 1] 0

Elemento[2, 2] 2

Elemento[2, 3] 0

Elemento[3, 1] 0

Elemento[3, 2] 0

Elemento[3, 3] 3

La matriz

	1	0	0	
	0	2	0	
	0	0	3	

es triangular INFERIOR y SUPERIOR.

Desea repetir el proceso con otros datos (S/N)?

DATOS DE LA MATRIZ

Elemento[1, 1] 9

Elemento[1, 2] 0

Elemento[1, 3] 0

Elemento[2, 1] 2

Elemento[2, 2] 3

Elemento[2, 3] 0

Elemento[3, 1] 4

Elemento[3, 2] 5

Elemento[3, 3] 6

La matriz

	9	0	0	
	2	3	0	
	4	5	6	

es triangular INFERIOR.

Desea repetir el proceso con otros datos (S/N)? s

DATOS DE LA MATRIZ

Elemento[1, 1] 5

Elemento[1, 2] 4

Elemento[1, 3] 3

Elemento[2, 1] 0

Elemento[2, 2] 2

Elemento[2, 3] 1

Elemento[3, 1] 0

Elemento[3, 2] 0

Elemento[3, 3] 0

La matriz

	5	4	3	
	0	2	1	
	0	0	0	

es triangular SUPERIOR.

Desea repetir el proceso con otros datos (S/N)? s

Ejercicio 2 (3 puntos)

En este ejercicio deberás:

- Implementar una función que calcule el máximo valor de todos lo que estén almacenados en un vector (**0,25 puntos**)
- Implementar una función que calcule el mínimo valor de todos lo que estén almacenados en un vector (**0,25 puntos**)
- Implementar una función que calcule la media de todos los valores que estén almacenados en un vector (**0,25 puntos**)
- Desarrollar el código que creas necesario en un programa escrito en C++11 para que a partir de los valores $\{\{1.1, 2.2, 3.3\}, \{3.3, 5.5\}, \{1.1, 1.1, 1.1, 1.1\}\}$ almacenados en una variable del tipo que consideres oportuno y haciendo uso de las funciones implementadas anteriormente sea capaz de calcular:
 - El valor máximo de los valores máximos (**0,75 puntos**). El resultado sobre los valores indicados en el enunciado debe dar 5.5
 - El valor medio de los valores máximos (**0,75 puntos**). El resultado sobre los valores indicados en el enunciado debe dar 3.3
 - El valor mínimo de las medias de los valores almacenados (**0,75 puntos**). El resultado sobre los valores indicados en el enunciado debe dar 1.1



Ejercicio 3 (4 puntos)

Diseñar e implementar un programa en C++11 que pida palabras al usuario por teclado. El usuario introduce palabra a palabra, y el programa va contando las veces que se ha introducido cada palabra, hasta que el usuario teclea “**salir**”.

Una vez que ha sido introducida la palabra “**salir**”, el programa imprimirá la **palabra que más aparece junto con el número de veces que aparece**, y la **palabra que menos veces aparece junto con su número de apariciones (2,5 puntos)**.

Seguidamente el programa imprimirá en **orden alfabético inverso** todas las palabras introducidas por el usuario (**1,5 puntos**).

Nota: Si las palabras se imprimen en orden alfabético normal, este apartado puntuará sobre **0,5 puntos** en lugar de 1,5 puntos, por lo que la nota máxima del ejercicio en ese caso sería 3 puntos en lugar de 4 puntos.

Por ejemplo:

casa

armario

silla

casa

salir

casa aparece mas veces 2

armario aparece menos veces 1

silla casa armario

En caso de que varias palabras aparezcan el mismo número de veces, se mostrará solo una de ellas como máximo o mínimo, en nuestro ejemplo *silla* y *armario* aparecen 1 vez, y solo se indica *armario* como mínimo. Por último, notad que *silla casa armario*, están en orden alfabético inverso y aunque *casa* se ha introducido dos veces solo aparece una.