

Indicaciones específicas:

- Esta evaluación contiene 8 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Debe crear un único proyecto CLION con archivos independientes por pregunta:
 - Pregunta1 → p1.h - p1.cpp
 - Pregunta2 → p2.h - p2.cpp
 - Pregunta3 → p3.h - p3.cpp
- El archivo main.cpp debe quedar así:

```
#include "p1.h"
#include "p2.h"
#include "p3.h"
int main() {
    pregunta1();
    pregunta2();
    pregunta3();
    return 0;
}
```

- Deberás subir un archivo ZIP que contenga únicamente los archivos *.h y *.cpp directamente a www.gradescope.com
- Recuerda que solo se calificará si has enviado en el formato indicado.

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
 - Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa. (nivel 2)
 - Diseñar, implementar y evaluar soluciones a problemas complejos de computación. (nivel 2)
 - Crear, seleccionar, adaptar y aplicar técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones. (nivel 2)
- Para los alumnos de las carreras de Ingeniería
 - Aplicar conocimientos de ingeniería en la solución de problemas complejos de ingeniería. (nivel 2)
 - Diseñar soluciones relacionados a problemas complejos de ingeniería. (nivel 2)

-
- Crear, seleccionar y utilizar técnicas, habilidades, recursos y herramientas modernas de la ingeniería y las tecnologías de la información, incluyendo la predicción y el modelamiento, con la comprensión de sus limitaciones. (nivel 2)
 - Para los alumnos de Administración y Negocios Digitales
 - Analizar información verbal y/o lógica proveniente de distintas fuentes, encontrando relaciones y presentándola de manera clara y concisa (nivel 2)
 - Analizar y evaluar el comportamiento del consumidor y el desarrollo de estrategias comerciales (nivel 2)
 - Trabajar de manera efectiva con equipos multidisciplinarios y diversos en género, nacionalidad, edad, etc. (nivel 2)
-

Calificación:

Tabla de puntos (solo para uso del profesor)

Question	Points	Score
<u>1</u>	7	
<u>2</u>	6	
<u>3</u>	7	
Total:	20	

1. (7 points) Generador de matrices cuadradas simétricas

Un poquito de teoría:

Una matriz cuadrada A es simétrica si y solo si es igual a su matriz transpuesta A^t
Por ejemplo:

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 3 & 0 & 5 \\ 1 & 5 & 1 \end{pmatrix} = A^t$$

Desarrolle un programa que permita realizar lo siguiente:

- Ingresar el tamaño de una matriz cuadrada validando que sea mayor a 2.
- Generar una **matriz dinámica simétrica** con valores aleatorios entre 1 y 100
- Mostrar en pantalla la matriz generada
- Liberar todo el espacio usado por la matriz.

Características mínimas a considerar en el desarrollo del programa:

- Uso de punteros para la generación de matrices dinámicas
- **Funciones para ingreso y validación de datos solicitados**
- **Funciones para la creación, generación, presentación y liberación de matrices dinámicas.**
- Usted debe decidir si las funciones requieren el uso de parámetros por valor, por referencia o por puntero.

Algunos ejemplos de dialogo de este programa serían:

Listing 1: Ejemplo 1

Ingrese dimesion de la matriz:

6

Matriz generada:

57 57 79 96 23 54

57 10 97 31 76 54

79 97 28 70 28 12

96 31 70 37 77 23

23 76 28 77 86 29

54 54 12 23 29 10

Listing 2: Ejemplo 1

```
Ingrese dimesion de la matriz:
```

```
3
```

```
Matriz generada:
```

```
63 31 16
```

```
31 94 21
```

```
16 21 62
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	El diseño tiene algunas deficiencias pero la ejecución es correcta (1pts).	El diseño es deficiente y la ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente(1pts).	El código no está optimizado y la ejecución es deficiente (0pts).

2. (6 points) Diferencia e Intersección de vectores

Desarrolle un programa que permita realizar lo siguiente:

- Generar dos vectores de n números enteros ($0 < n \leq 10$). Cada vector deberá tener **números aleatorios no repetidos** entre 1 y 20.
- Mostrar ambos vectores con la notación de la teoría de conjuntos ($A=\{\}$, $B=\{\}$)
- Crear y mostrar un tercer vector que resulte de realizar la operación de diferencia de conjuntos ($A-B$)
- Crear y mostrar un cuarto vector que resulte de realizar la operación de intersección de conjuntos ($A \cap B$)

Características para considerar en el desarrollo del programa:

- Uso del tipo vector como contenedor de datos.
- **Funciones para ingreso y validación de datos solicitados**
- **Funciones para la creación, generación, presentación de los vectores que representan a los conjuntos.**
- **Función para realizar la diferencia de conjuntos**
- **Función para realizar la intersección de conjuntos**
- Usted debe decidir si las funciones requieren el uso de parámetros por valor, por referencia o por puntero.

Algunos ejemplos de dialogo de este programa serían:

Listing 3: Ejemplo 1

```
Numero de elementos:
5
A={14,19,3,6,18}
B={13,7,2,14,4}
A-B={19,3,6,18}
A∩B={14}
```

Listing 4: Ejemplo 1

```
Numero de elementos:
3
A={17,19,2}
B={12,1,17}
A-B={19,2}
A∩B={17}
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (2pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (1.5pts)	El diseño tiene algunas deficiencias pero la ejecución es correcta (1pts).	El diseño es deficiente y la ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente(1pts).	El código no está optimizado y la ejecución es deficiente (0pts).

3. (7 points) Clase Calculadora Estadística

Un poquito de teoría:

Media

La media de un conjunto de números, algunas ocasiones simplemente llamadas **el promedio**, es la suma de los datos dividida entre el número total de datos.

Mediana

La mediana de un conjunto de números es el número medio en el conjunto **(después que los números han sido ordenados de menor al mayor)** o, si hay un número par de datos, la mediana es el promedio de los dos números medios.

Desarrolle un POO que contenga una clase denominada **Estadística** con las siguientes características:

- Que en su constructor reciba un vector de números double conteniendo los datos de una muestra numérica.
- Que tenga los siguientes métodos de comportamiento:
 - hallarMedia(), el cual deberá calcular el promedio.
 - hallarMediana(), el cual deberá calcular la mediana.
 - mostrarDatos(), el cual deberá presentar en pantalla los datos de la muestra ordenada de mayor a menor y con el formato {dato1, dato2, dato3,...}.
- Que en su destructor libere los datos de la muestra

Pruebe la funcionalidad de su clase, haciendo cumplir el ciclo de vida los objetos: crear, usar y destruir.

Use las siguientes muestras: {18, 20, 23, 22, 21} y {3, 10, 36, 80, 79, 24, 5, 8}

Algunos ejemplos de dialogo de este programa serían:

Listing 5: Ejemplo 1

```
Muestra: {18,20,21,22,23}  
Media: 20.8  
Mediana: 21
```

Listing 6: Ejemplo 1

```
Muestra: {3,5,8,10,24,36,79,80}  
Media: 30.625  
Mediana: 17
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	El diseño tiene algunas deficiencias pero la ejecución es correcta (1pts).	El diseño es deficiente y la ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente(1pts).	El código no está optimizado y la ejecución es deficiente (0pts).