UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II
Práctica Calificada 2
Pregrado
2022-I
Profesor: Estanislao Contreras
Lab 1.02

Indicaciones específicas:

- Esta evaluación contiene 9 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
1	7	
2	6	
3	7	
Total:	20	

1. **(7 points) Transposición de Matrices**

Desarrolle un programa que permita realizar lo siguiente:

- Ingresar el tamaño de una matriz validando que filas y columnas sean mayor a 2.
- Generar una **matriz dinámica** con valores aleatorios entre 1 y 1000
- Mostrar en pantalla la matriz generada
- Generar una **segunda matriz dinámica** como la transpuesta de la primera. Recuerde que la matriz transpuesta se obtiene cambiando las filas por columnas o viceversa.
- Mostrar en pantalla la matriz transpuesta.
- Liberar todo el espacio usado por ambas matrices.

Características mínimas para 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.**
- **Función para la transposición**
- 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

```
Ingresar numero de filas:
4
Ingresar numero de columnas:
5
Matriz generada:
469 445 242 92 402
911 448 803 731 83
676 72 153 41 331
568 512 851 919 673
-----------------------------------------
Matriz transpuesta:
469 911 676 568
445 448 72 512
242 803 153 851
92 731 41 919
402 83 331 673
-----------------------------------------
```

Listing 2: Ejemplo 1

```
Ingresar numero de filas:
3
Ingresar numero de columnas:
3
Matriz generada:
759  13  289
39  895  942
545  419  703
-----------------------------------------
Matriz transpuesta:
759  39  545
13  895  419
289  942  703
-----------------------------------------
```

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) Unión e Intersección de vectores

Desarrolle un programa que permita realizar lo siguiente:

- Generar dos vectores de n números enteros (0<n<=10). Cada vector deberá tener **números aleatorios no repetidos** entre 1 y 20.
- Mostrar ambos vectores
- Crear y mostrar un tercer vector que resulte de realizar la unión de los elementos contenidos en los 2 primeros vectores.
- Crear y mostrar un cuarto vector que resulte de realizar la intersección de los elementos contenidos en los 2 primeros vectores

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.**
- **Función para realizar la unión de 2 vectores**
- **Función para realizar la intersección de 2 vectores**
- 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
Primer Vector
[19 10 6 14 18 ]
Segundo Vector
[3 4 6 16 19 ]
Union
[19 10 6 14 18 3 4 16 ]
Interseccion
[19 6 ]
```

Listing 4: Ejemplo 1

```
Numero de elementos:
3
Primer Vector
[14 8 16 ]
Segundo Vector
[7 2 6 ]
Union
[14 8 16 7 2 6 ]
Interseccion
[]
```

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 Estadístisca

Desarrolle un POO que contenga una clase denominada **Estadistica** 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:
  - hallarMaximo(), el cual deberá hallar el dato mayor.
  - hallarMinimo(), el cual deberá hallar el dato menor.
  - hallarMedia(), el cual deberá calcular el promedio.
  - mostrarDatos(), el cual deberá presentar en pantalla los datos de la muestra.
- 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.

Uses el siguiente set de datos de prueba: 19 20 23 22 21

Algunos ejemplos de dialogo de este programa serían:

Listing 5: Ejemplo 1

```
[19 20 23 22 21]
Mayor : 23
Menor : 19
Media : 21
```

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).