UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II
Práctica Calificada 2
Pregrado
2022-II
Profesor: José A. Chávez Álvarez
Lab 2.01

Indicaciones específicas:

- Esta evaluación contiene 7 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
  - `p1.cpp`
  - `p2.cpp`
  - `p3.cpp`
- Deberás subir estos archivos directamente a www.gradescope.com, uno en cada ejercicio. También puedes crear un .zip

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación

  Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)

  Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)

  Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)

- Para los alumnos de las carreras de Ingeniería

  Capacidad de aplicar conocimientos de matemáticas (nivel 3)

  Capacidad de aplicar conocimientos de ingeniería(nivel 2)

  Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	7	
2	6	
3	7	
Total:	20	

1. (7 points) **Matrices con Punteros**
   Implemente un programa que genere matrices dinámicas con valores aleatorios. El programa debe tener un menú con tres opciones:

   - Opción 1: Generar una matriz con números enteros aleatorios entre 1 y 99. En esta opción se debe solicitar la cantidad de filas y columnas. Se debería poder crear varias veces una matriz.
   - Opción 2: Detectar los elementos que son múltiplos de tres.
   - Opción 3: Salir del menú.

   Un ejemplo del diálogo de este programa sería:

Listing 1: Ejemplo 1

```
MENU:
1. Generar matriz aleatoria con numeros entre 1 y 99.
2. Detectar los multiplos de 3.
3. Salir.

Opcion: 1
#Filas: 4
#Columnas: 5

 4 15 32 99 12
 2  1  1 15 23
43 12 11 10  9
13 77 88 51  2

MENU:
1. Generar matriz aleatoria con numeros entre 1 y 99.
2. Detectar los multiplos de 3.
3. Salir.

Opcion: 2

 0 15  0 99 12
 0  0  0 15  0
 0 12  0  0  9
 0  0  0 51  0

MENU:
1. Generar matriz aleatoria con numeros entre 1 y 99.
2. Detectar los multiplos de 3.
3. Salir.

Opcion: 1
```

```
#Filas: 3
#Columnas: 3

14 17 33
 3 11 21
44 17 91

MENU:
1. Generar matriz aleatoria con numeros entre 1 y 99.
2. Detectar los multiplos de 3.
3. Salir.

Opcion: 3
Fin.
```

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)	La ejecución es correcta (1pts).	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 sintáxis que afectan el resultado (0.5pts).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El codigo es de buen performance durante la ejecución (1.5pts).	El codigo no está optimizado pero la ejecución no es deficiente(1pts).	El codigo no está optimizado y la ejecución es deficiente (0pts).

2. (6 points) **Vectores**

Generar un vector con 20 números enteros aleatorios entre 1 y 99, luego calcule:

- El promedio de todos los números del vector.
- Genere dos vectores a partir de los elementos del vector original:
  - El primer vector con todos los números pares.
  - El segundo vector con todos los números impares.
- Genere una función para imprimir en una linea cada vector.

Algunos ejemplos de diálogo de este programa serían:

Listing 2: Ejemplo 2

```
Vector 1: 12 15 21 27 77 85 99 1 15 35 22 48 76 35 69 24 21
   36 11 12
Vector 2: 12 22 48 76 24 36 12
Vector 3: 26 21 27 77 85 99  1 15 35 35 69 21 11
```

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`)	La ejecución es correcta (`1pts`).	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 sintáxis que afectan el resultado (`0.5pts`).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (`2pts`)	El codigo es de buen performance durante la ejecución (`1.5pts`).	El codigo no está optimizado pero la ejecución no es deficiente(`1pts`).	El codigo no está optimizado y la ejecución es deficiente (`0pts`).

3. (7 points) **Clases y Objetos**
Implementar la clase `Curso` con los siguientes caracteristicas:

   - Dentro del constructor se debe se debe configurar el nombre del curso, ciclo y créditos.
   - El destructor debe imprimir la palabra `Fin`.
   - SETTERS
     - `setAlumno`: permite ingresar el nombre del alumno (`nombre`) y su nota final (`nota`).
   - GETTERS:
     - `getProm`: permite obtener el promedio de notas finales de todos los alumnos.
   - El método `printRecord` debe permitir visualizar el récord de notas de todos los alumnos. Imprimir el nombre y nota para cada alumno.
   - Implementar la clase `Curso` dentro del archivo `p3.cpp`.

   Completar el siguiente código (`p3.cpp`)

Listing 3: Ejemplo 3

```
/* Implementar la Clase aqui
 * ==========================
 *
 * ==========================
*/

int main(){
    Curso obj("Prog2", 2, 4);

    obj.setAlumno("Juan", 14);
    obj.setAlumno("Maria", 13);
    obj.setAlumno("Miguel", 11);
    obj.setAlumno("Elena", 17);

    cout << "El promedio es: " << obj.getProm() << endl;
    cout << endl;
    obj.printRecord();
    cout << endl;
}
```

El resultado debería ser el siguiente:

Listing 4: Ejemplo 4

```
El promedio es: 13.75

Curso : Prog2
Ciclo : 2
Creditos: 4

Alumno   Nota
Juan    -> 14
Maria   -> 13
Miguel  -> 11
Elena   -> 17

Fin.
```

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)	La ejecución es correcta (1pts).	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 sintáxis que afectan el resultado (0.5pts).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El codigo es de buen performance durante la ejecución (1.5pts).	El codigo no está optimizado pero la ejecución no es deficiente(1pts).	El codigo no está optimizado y la ejecución es deficiente (0pts).