UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II
Práctica Calificada 2
Pregrado
2022-I
Profesor: Jaime Farfán
Lab: 2.03

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 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 (sólo para uso del professor)

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

1. (7 points) **MATRICES CON PUNTEROS**
Pedir desde teclado los valores N(FILAS) y M(COLUMNAS), crear una matriz dinámica (punteros). Asignar valores aleatorios a la matriz en el rango de 10-99.

   - **A las columnas PARES RESTAR el valor MÍNIMO de la matriz.**

   Formatear la matriz resultado para que los valores se encuentren alineados.

   Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado

   Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Ingrese N : 4
Ingrese M : 3

Minimo: 20

MATRIZ GENERADA
32 53 20
25 94 34
57 36 35
25 26 90

MATRIZ RESULTADO
32 33 20
25 74 34
57 16 35
25 06 90
```

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 sintáxis que afectan el resultado** (0.5pts)
Optimizacion	**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) VECTORES
Crear 2 vectores de 10 elementos cada uno, asignar números aleatorios a cada vector en el rango de 1-99.

- **Colocar 00, en el vector2 a los números que se encuentran en el vector1.**

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado

Listing 2: Ejemplo 1

```
Valores generados:
vector1: 67 83 52 38 24 11 17 94 29 16
vector2: 11 46 43 68 44 56 24 68 75 84

resultado:
vector2: 00 46 43 68 44 56 00 68 75 84
```

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 sintáxis que afectan el resultado** (0.5pts)
Optimizacion	**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) CLASES Y OBJETOS
Crear la clase PRODUCTO con los campos codigo, nombre, precio,categoria.

- **Crear la clase con los atributos indicados.**
- **Generar setter y getter.**
- **Crear el constructor vacío y otro con todos los parámetros**
- **Validar que el precio acepte valores a 0, en caso contrario enviar mensaje FUERA DE RANGO**
- **Validar que la categoria acepte solo valores [A,B,C,D], en caso contrario enviar mensaje FUERA DE RANGO**

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado

Listing 3: Ejemplo 1

```
Producto 1:
codigo: 2114
nombre: Monitor LG
precio: 3500
categoria: A

RESULTADO:
producto 1 : Codigo: 2114, Nombre: Monitor LG, Precio: 3500,
    Categoria: A
```

Listing 4: Ejemplo 1

```
Producto 2:
codigo: 2114
nombre: Monitor LG
precio: 0
categoria: A

RESULTADO:
producto 2 : Codigo: 2114, Nombre: Monitor LG, Precio: FUERA
    DE RANGO, Categoria: A
```

Listing 5: Ejemplo 1

```
Producto 3:
codigo: 2114
nombre: Monitor LG
precio: 2800
categoria: G
```

```
RESULTADO:
producto 3 : Codigo: 2114, Nombre: Monitor LG, Precio: 2800,
    Categoria: FUERA DE RANGO
```

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 sintáxis que afectan el resultado** (0.5pts)
Optimizacion	**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)