

### 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](http://www.gradescope.com), uno en cada ejercicio. También puedes crear un .zip
- **No está permitido** copiar soluciones de fuentes externas. Si se detecta entregas con soluciones comunes, se anulará la prueba.
- Se solicita activar cámara durante la evaluación. En caso no ser posible, justifique por correo electrónico a [jfiestas@utec.edu.pe](mailto:jfiestas@utec.edu.pe)

### 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) **Procesamiento gráfico en dos dimensiones**

Una matriz puede representar una imagen, que está sujeta a operaciones de procesamiento gráfico, como las siguientes:

- **Intensidad:** duplica la intensidad de cada pixel con valor menor a 50, y reduce la intensidad de pixeles con valores mayores a 50 con un factor 0.5
- **Suavizado** de cada elemento. Lo que consiste en promediar su valor  $(x_{i,j})$  con los valores vecinos laterales, es decir  $x_{ij} = \frac{x_{i-1,j} + x_{i,j} + x_{i+1,j}}{3}$ . El suavizado se repite una cantidad  $n$  de veces que el usuario ingrese. Note que los valores en los extremos de la imagen no tienen vecinos en todas las direcciones. Adapte la fórmula en estos pixeles.
- **Rotacion** de sus elementos en un ángulo  $\theta$ , aplicando la siguiente fórmula de rotación a la posición de cada pixel de la imagen

$$i' = i \cos(\theta) - j \sin(\theta)$$

$$j' = i \sin(\theta) + j \cos(\theta)$$

Utilice  $\theta = 90^\circ$ . Note que, ya que  $\sin(90)=1$  y  $\cos(90)=0$ , al rotar la imagen  $90^\circ$ , los pixeles pueden tener posiciones negativas, por lo que debe realizar una translación de los pixeles para que las posiciones  $(i,j)$  empiecen en  $(0,0)$

La imagen tiene inicialmente un tamaño de  $600 \times 900$  pixels, e intensidades generadas aleatoriamente entre 0 y 100. Ejecute cada operacion e imprima el promedio de los valores de los pixeles de la imagen luego de cada una de las 3 operaciones.

Utilice matrices dinámicas para resolver el ejercicio.

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

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 pts)
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)
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) **Norma-p**

Programe el cálculo de la norma-p en un espacio n-dimensional, de acuerdo a

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$$

$$\|x\|_\infty = \max_{i=1}^n |x_i|$$

Donde  $n, p \geq 1$ ,  $n, p \in \mathbb{N}$ , max se refiere al elemento del vector con el mayor valor absoluto.

Para ello:

- Genere un vector de floats, de n elementos (input), asignando valores aleatorios para cada componente, tal que  $0 < x_i < 100$ , y  $\vec{x} = (x_0, \dots, x_n)$
- Calcule e imprima la norma  $\|x\|_1$ ,  $\|x\|_2$ ,  $\|x\|_\infty$
- Calcule e imprima los vectores  $x/\|x\|$ , es decir, con elementos  $\frac{x_i}{1\text{-norm}}$ ,  $\frac{x_i}{2\text{-norm}}$ ,  $\frac{x_i}{\text{max-norm}}$

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

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 pts)
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)
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) **números binarios de 64 bits**

Defina la clase **NumeroBinario** que represente números binarios de 64 bits, por ejemplo con un vector de dimension 64. La clase debe contener los siguientes métodos:

- a) Constructor, que inicializa todos los bits a cero
- b) método que imprima el valor del número binario, con el bit de mayor valor primero, y el de menor valor al final.
- c) Un método que pida un numero entero y permita asignar el valor a un bit a 1, en esa posición. La función retorna el valor **true** si el valor del bit es 1 o si se cambio a 1, y **false** en caso contrario (no se pudo cambiar)
- d) método que permita incrementar el valor de **un bit** de 0 a 1. Para ello se empieza a recorrer los bits de menor a mayor (derecha a izquierda). Si el bit es 0, se convierte a 1, y termina la ejecución. Si el bit es 1, se convierte a 0, y se sigue con el siguiente bit.

El programa debe ejecutar la siguiente secuencia:

- imprimir el valor inicial del número binario
- usar el método c) para pedir un numero n e intentar asignar el bit a 1 en esa posicion
- fijar en 1 los bits en las posiciones 2 a 20, 61 y 62
- imprimir el número binario
- llamar el método d) 50 veces e imprimir el número binario

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

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 pts)
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)
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)