

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
- **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

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) Población de garzas

La población de garzas en Chorrillos es de 300. Inicialmente están distribuidas en 100 polluelos, 100 garzas jóvenes y 100 adultas. Es decir, representables en un array $x = [100, 100, 100]$. Su reproducción se rige por los siguientes parámetros

- cada año, nace por garza adulta, un porcentaje **A** de polluelos
- cada año, sobrevive un porcentaje **D** de garzas adultas
- cada año, sobrevive un porcentaje **B** de polluelos y se convierte en garzas jóvenes
- cada año, sobrevive un porcentaje **C** de garzas jóvenes y se convierte en garzas adultas

donde $A, B, C, D \in [0, 1]$

De la población $x = [100, 100, 100]$ se puede obtener la población $y = [y_1, y_2, y_3]$ cada año, al multiplicar x con la matriz

$$\begin{pmatrix} 0 & 0 & a \\ b & 0 & 0 \\ 0 & c & d \end{pmatrix}$$

Recuerde que la multiplicación matriz-vector se define como

$$\begin{pmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{pmatrix} \times \begin{pmatrix} x_{00} \\ x_{10} \\ x_{20} \end{pmatrix} = \begin{pmatrix} m_{00}x_{00} + m_{01}x_{10} + m_{02}x_{20} \\ m_{10}x_{00} + m_{11}x_{10} + m_{12}x_{20} \\ m_{20}x_{00} + m_{21}x_{10} + m_{22}x_{20} \end{pmatrix}$$

Utilice valores para $a=0.5$, $b=0.5$, $c=0.5$, $d=0.9$, y escriba un código que permita ingresar la cantidad de años (n) y calcule las poblaciones x^n

Ahora utilice $a=0.5$, $b=0.5$, $c=0.5$, $d=0.5$, y compare el resultado con el caso anterior, para el mismo valor de n .

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) **Suavizado en una dimensión**

Aplique las siguientes operaciones a los elementos de un vector que consiste inicialmente de 30 números enteros, aleatorios entre 4 y 88

- ordenamiento de los elementos del vector
- desplazamiento un elemento a la derecha y borrado del primer elemento. Por ejemplo: el vector $\{2,3,5,1,7,5,8,9\}$, luego del desplazamiento, queda $\{2,3,5,1,7,5,8\}$
- desplazamiento un elemento a la izquierda y borrado del último elemento. Por ejemplo: el vector $\{2,3,5,1,7,5,8,9\}$, luego del desplazamiento, queda $\{3,5,1,7,5,8,9\}$
- suavizado de cada elemento. Lo que consiste en promediar su valor con los valores vecinos, es decir $x_i = \frac{x_{i-1} + x_i + x_{i+1}}{3}$. Se repetirá el suavizado una cantidad n de veces que el usuario ingrese.

Imprima el vector resultante luego de cada operación.

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) **Inmuebles**

Crear una clase **Inmueble** con atributos para el **área**, y **precio por metro cuadrado**, así como 3 constructores para la clase.

Genere un procedimiento de compra/venta. Para ello, programe un **menú** que permita comprar un inmueble (condominio, casa u hotel). Realice la compra de los siguientes inmuebles:

- **3 departamentos**, cada uno con un precio por metro cuadrado aleatorio entre 5000 y 10000 (Soles), y area aleatoria entre 300 y 500 (m^2)
- **4 casas**, cada uno con un precio por metro cuadrado aleatorio entre 10000 y 15000 (Soles), y area aleatoria entre 500 y 1000 (m^2)
- **2 hoteles**, cada uno con un precio por metro cuadrado aleatorio entre 15000 y 30000 (Soles), y area aleatoria entre 1000 y 1500 (m^2)

El pago será a plazos, para lo que es necesario calcular el **tiempo de pago** si se decide por un pago mensual, que debe ser dado por el usuario (input), si se tiene un interés del 0.5% mensual.

Almacene los datos de los inmuebles comprados en un **array dinámico** o **vector**, y genere una **función externa** a la clase para imprimir, en cada caso, el nombre del inmueble, su área, precio por metro cuadrado, y el tiempo de pago (en años). Asimismo, debe poder calcular el saldo de la deuda después de una cantidad determinada de meses, que el usuario ingrese. Utilice punteros a objetos y comente por qué los usa en este código.

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)