



Nombre: _____ fecha: 05/03/2024

Reglas de la prueba

- **Comunicación:** Está prohibido hablar o comunicarse de cualquier forma con otros estudiantes durante el examen.
- **Integridad Académica:** Cualquier forma de trampa, incluido el plagio, copiado o uso de material no autorizado, resultará en una calificación de cero en el examen y puede llevar a sanciones adicionales según las políticas de RIWI.
- **Material Permitido:** Solo se permite ver material de apoyo como lo son diapositivas o ejercicios realizados en clase, si se requiere ver estos materiales de apoyo se debe comunicar al Trainer y se tendrá un máximo de 5 minutos para revisar el material y con un máximo de 3 veces por estudiantes, el Trainer debe estar presente con el coder supervisando que se esté observando el material indicado.
- **Permanencia en el Aula:** Una vez iniciado el examen, no se permite salir del aula hasta haber entregado el examen y, de preferencia, hasta que haya transcurrido al menos la mitad del tiempo asignado.
- **Entrega:** Una vez finalizada la prueba se debe enviar en una carpeta comprimida con el nombre y apellidos del coder al correo electrónico, en la carpeta comprimida deben estar los tres ejercicios, cada uno en un archivo independiente.

Problema 1

Enunciado:

Escriba un programa en Python que solicite al usuario que calcule la suma y la cantidad de los números pares que hay en la siguiente lista:

[55, 84, 257, 954, 2211, 356, 17, 14, 2547, 8452]

Requerimientos:

El programa debe solicitar al usuario que ingrese una lista de números.

El programa debe permitir al usuario ingresar la lista de números, ya sea separados por comas, espacios u otro delimitador.

El programa debe identificar el número más grande de la lista ingresada por el usuario.

El programa debe identificar el número más pequeño de la lista ingresada por el usuario.

El programa debe calcular el promedio de los números de la lista ingresada por el usuario.

El programa debe mostrar al usuario el número más grande, el número más pequeño y el promedio de los números de la lista.



El programa debe estar escrito en el lenguaje de programación Python.

Problema 2

Enunciado:

Descripción del Problema

Imagina que una compañía de seguros ofrece pólizas de seguro de vida con primas variables, que dependen de la edad del asegurado. Si la edad está entre 18 y 30 años, la prima es del 5%. Para edades entre 31 y 50 años, la prima es del 8%. Si la edad supera los 50 años, la prima es del 12%. Desarrolla un programa que calcule el total de primas generadas por las diferentes personas que han adquirido estas pólizas de seguro de vida.

Requerimientos

El programa debe permitir al usuario ingresar la edad de cada asegurado. El ingreso de datos debe continuar hasta que el usuario decida finalizar.

Para cada edad ingresada, el programa debe determinar la prima correspondiente según las siguientes condiciones:

- Si la edad está entre 18 y 30 años, la prima es del 5%.
- Para edades entre 31 y 50 años, la prima es del 8%.
- Si la edad supera los 50 años, la prima es del 12%.

El programa debe calcular y mantener un registro de la suma total de todas las primas generadas por los asegurados.

- Visualización del Total de Primas: Al finalizar el ingreso de datos, el programa debe mostrar el total de primas generadas por todas las personas aseguradas.

- Implementación en Python: El programa debe estar escrito en el lenguaje de programación Python.

Problema 3

Enunciado:

Un Zoólogo pretende determinar el porcentaje de animales con alguna patología que los afecta que hay en las siguientes tres categorías de edades: de 0 a 1 año, de más de 1 año y menos de 3 y de 3 o más años. El zoológico todavía no está seguro del animal que va a estudiar. Si se decide por elefantes solo tomara una muestra de 20 de ellos; si se decide por las jirafas, tomara 15 muestras, y si son chimpancés tomara 40.

Su misión es hacer un programa que calcule por especie el porcentaje de animales que tiene o no la patología por especie en cada categoría.



Requerimientos:

El programa debe permitir al usuario seleccionar la especie de animal a estudiar entre elefantes, jirafas y chimpancés.

Si el usuario elige elefantes, el programa debe tomar una muestra de 20 elefantes para el estudio.

Si elige jirafas, el programa debe tomar una muestra de 15 jirafas.

Si elige chimpancés, el programa debe tomar una muestra de 40 chimpancés.

El programa debe solicitar al usuario ingresar el número de animales con patología para cada categoría de edad:

- Categoría 1: de 0 a 1 año.
- Categoría 2: más de 1 año y menos de 3.
- Categoría 3: 3 o más años.

Para cada categoría de edad, el programa debe calcular y mostrar el porcentaje de animales con o sin patología.

Rubrica de Evaluación:

Criterio	Puntuación 3 (Excelente)	Puntuación 2 (Bueno)	Puntuación 1 (Necesita Mejorar)	Puntuación 0 (no presenta)
Conocimiento del Anunciado	El estudiante muestra entendimiento del enunciado y cumple con todos los requisitos de manera.	El estudiante demuestra comprensión del enunciado y cumple con la mayoría de los requisitos.	El estudiante tiene dificultades para entender el enunciado y no cumple con los requisitos clave.	El estudiante no demuestra entender ningún enunciado
Calidad del Código	El código del estudiante es limpio, bien estructurado y eficiente. Se siguen las mejores prácticas de programación.	El código del estudiante es generalmente claro y funcional, pero podría haber mejoras en la estructura.	El código del estudiante es desorganizado, difícil de leer y contiene errores graves.	La cantidad de código escrito, no es suficiente para determinar su calidad



Cumplimiento de Requisitos Técnicos	El estudiante cumple con todos los requisitos técnicos del ejercicio de manera precisa y completa.	El estudiante cumple con la mayoría de los requisitos técnicos, pero puede haber omisiones o errores menores.	El estudiante no logra cumplir con varios requisitos técnicos clave.	El estudiante no cumple con ningún requisito técnico
Comentarios y Documentación del Código	El estudiante proporciona comentarios detallados y documentación clara en el código. Los comentarios explican de manera efectiva el propósito y la lógica del código.	El estudiante proporciona comentarios adecuados, pero puede haber algunas áreas que carecen de documentación.	El código del estudiante carece de comentarios o documentación significativa.	El código no evidencia ningún comentario o documentación
Sustentación	El estudiante muestra un profundo entendimiento del código y demuestra conocimiento completo sobre la lógica, estructura, funciones y métodos	El estudiante demuestra comprensión del código, y la lógica, pero no identifica estructuras, funciones o métodos.	El estudiante tiene dificultades para entender el enunciado no identifica la lógica del código	El estudiante no presenta sustentación