



UNIVERSIDAD ARGENTINA DE LA EMPRESA

Departamento de Tecnología Informática

INTRODUCCIÓN A LA ALGORITMIA

Profesor: Manuel Adrian Caceres

Examen Final Adelantado

ALUMNO:

LU:

FECHA: 29/11/2024

CARRERA: Licenciatura en Gestión de Tecnología de la Información

TEMA: 1

NOTA: EL EXAMEN ESCRITO ES UN DOCUMENTO DE GRAN IMPORTANCIA PARA LA EVALUACIÓN DE LOS CONOCIMIENTOS ADQUIRIDOS, POR LO TANTO, SE SOLICITA LEER ATENTAMENTE LO SIGUIENTE:

- Responda claramente cada punto, detallando con la mayor precisión posible lo solicitado.
- Sea prolijo y ordenado en el desarrollo de los temas.
- Sea cuidadoso con las faltas de ortografía y sus oraciones.
- No desarrollar el examen en lápiz.
- Aprobación del examen: Con nota mayor o igual a 4 (cuatro).
- Entrega del Examen: cargar en teams los archivos generados en python

TEÓRICOS						PRÁCTICOS			
1	2	3	4	5	6	7	8	9	10
NOTA EN NUMEROS		NOTA EN LETRAS		SELLO		FIRMA DEL DOCENTE			

Ejercicio 1 (1,5 puntos + 0,5 puntos usando funciones con parámetros):
Análisis de Ventas Diarias:

Desarrolla un programa en Python que simule la generación de una lista de 30 números enteros aleatorios, cada uno representando ventas diarias en un rango entre 50 y 500 unidades.

El programa debe:

Imprimir todos los días en que las ventas estuvieron por debajo de 150 unidades.
 Calcular e imprimir la cantidad promedio de ventas (sin usar funciones predefinidas como sum o len).

Restricciones:

No se permite el uso de las funciones predefinidas de Python como min, max, sum, len ni listas por comprensión.
 La lógica debe ser implementada manualmente.

Ejercicio 2 (2 puntos + 0,5 por uso de funciones con parámetros + 0,5 por implementación de ordenamiento por selección):**Filtrar y Ordenar Animales por Longitud del Nombre**

Crear una lista de al menos 15 nombres de animales. El programa debe:

1. Seleccionar los nombres que tengan más de 8 caracteres.
2. Guardar estos nombres en una nueva lista.
3. Ordenar la nueva lista en orden alfabético utilizando el algoritmo de ordenamiento por selección (Selection Sort).
4. Mostrar la nueva lista ordenada.

Restricciones:

- La selección de nombres y el ordenamiento deben implementarse manualmente.
- No se permite el uso de métodos predefinidos como sort o sorted.

Ejercicio 3 (1,5 puntos + 0,5 usando funciones con parámetros)**Operaciones con Matrices y Diagonales**

- Generar dos matrices de 5 filas y 5 columnas, llenando cada una con números aleatorios entre 10 y 50.
- Sumar los valores correspondientes de las dos matrices, celda por celda, ejemplo `matriz1[4][4]+matriz2[4][4]` y guardar el resultado en una tercera matriz.
- Calcular el promedio de los elementos de la diagonal principal de la tercera matriz.
- Mostrar las tres matrices (las dos originales y la resultante) y el promedio calculado.

Restricciones:

- Usa funciones para separar la lógica de generación de matrices, suma de matrices y cálculo del promedio de la diagonal.
- Evita usar librerías como numpy para garantizar la implementación manual de la lógica.

Ejercicio 4 (2 puntos + 0,5 usando funciones con parámetros + 0,5 punto por uso de algoritmo de ordenamiento de selección)**Gestión de Productos y Precios de Venta**

Crear una matriz con datos de prueba que contenga:

Columna 1: ID del producto.

Columna 2: Nombre del producto.

Columna 3: Precio de compra.

Calcular el precio de venta para cada producto, incrementando el precio de compra en un 20%.

Crear una nueva matriz igual a la anterior, pero agregando una cuarta columna con el precio de venta. Ordenar la nueva matriz por la columna del precio de venta, utilizando el algoritmo de burbuja (Bubble Sort).

Imprimir ambas matrices (la original y la nueva ordenada).