



Taller 01

Análisis y Diseño de Algoritmos

Instrucciones

- Resuelve todos los ejercicios utilizando el lenguaje de programación **Rust**.
- El trabajo debe realizarse en grupos de hasta **tres integrantes**.
- Cada grupo deberá subir al sistema:
 - Un archivo en formato **PDF** con el desarrollo de los ejercicios.
 - Los **códigos fuente** correspondientes (obligatorio).

Problemas

Problema 1: Promedio de Lista Aleatoria (1 punto).

Crea una función que genere una lista aleatoria de N calificaciones con valores entre 0 y 20, y calcule el promedio, la mediana y la moda. Si hay menos de 3 notas, retorna **None**.

Entrada (se debe ingresar desde consola):

Ingrese N: 7

Salida esperada:

Lista generada: [14, 17, 18, 12, 19, 14, 13]

Promedio: 15.29

Mediana: 14

Moda: 14

Problema 2: Diagonal Secundaria (2 puntos).

Crea una función que genere una matriz cuadrada $N \times N$ de elementos aleatorios y que retorne:

- La matriz $N \times N$ generada.
- Los elementos de la diagonal secundaria (de derecha a izquierda, desde arriba).
- La suma de los elementos de dicha diagonal.

Entrada (se debe ingresar desde consola):

Ingrese N: 3

Salida:

Matriz generada: `[[5, 1, 9], [2, 8, 3], [4, 6, 7]]`

Diagonal secundaria: `[9, 8, 4]`

Suma de diagonal: `21`

Problema 3: Transposición de Matriz (1 punto).

Implementa una función que reciba una matriz de tamaño $m \times n$ y retorne su transpuesta ($n \times m$). Si la matriz es cuadrada, además debe retornar la suma de los elementos de la diagonal principal.

Problema 4: Contar celdas mayores que el promedio (2 puntos).

Genera una matriz $M \times N$ con valores enteros aleatorios entre 10 y 100. Calcula el promedio de todos los elementos y retorna:

- La matriz generada,
- El promedio,
- Cuántos elementos son estrictamente mayores que el promedio.

Ejemplo:

```
50  70  90
40  60  80
```

Salida esperada:

Promedio 65:

3 celdas son mayores que el promedio

Problema 5: Secuencia Zig-Zag (2 puntos).

Escribe una función que determine si una lista de enteros forma una secuencia zig-zag; es decir, los elementos deben alternar entre subir y bajar.

Ejemplo:

```
lista = [1, 3, 2, 4, 3]
```

Salida:

¿Es Zig-Zag? `True`

Problema 6: Rango de aparición de un número en una lista ordenada (2 puntos).

Dada una lista ordenada de enteros y un número objetivo, devuelve el primer y último índice donde aparece ese número. Si no aparece, retorna $(-1, -1)$.

Ejemplo:

```
lista = [1, 2, 2, 2, 3, 4, 5]
objetivo = 2
```

Salida:

```
(1, 3)
```

Problema 7: Verificador de Sudoku Parcial (3 punto).

Escribe una función que reciba una matriz 9×9 (matriz de enteros de 0 a 9, donde 0 significa celda vacía) la matriz debe estar definida en el código (no pedir que se ingrese), y determine si no hay repeticiones de números (del 1 al 9) en filas, columnas y subcuadrantes 3×3 llenos hasta el momento.

Matriz:

```
5 3 0 | 0 7 0 | 0 0 0
6 0 0 | 1 9 5 | 0 0 0
0 9 8 | 0 0 0 | 0 6 0
-----+-----+-----
8 0 0 | 0 6 0 | 0 0 3
4 0 0 | 8 0 3 | 0 0 1
7 0 0 | 0 2 0 | 0 0 6
-----+-----+-----
0 6 0 | 0 0 0 | 2 8 0
0 0 0 | 4 1 9 | 0 0 5
0 0 0 | 0 8 0 | 0 7 9
```

Salida esperada:

```
El tablero es válido.
```

Problema 8: Agrupar números consecutivos (2.5 puntos).

Dada una lista de enteros ordenada, implementa una función que agrupe los números consecutivos en sublistas.

Ejemplo:

```
lista = [1, 2, 3, 5, 6, 9]
```

Salida esperada:

```
[[1, 2, 3], [5, 6], [9]]
```

Problema 9: Evaluación de Matriz Triangular Superior (2.5 puntos).

Escriba un programa que determine si una matriz cuadrada es triangular superior. Una matriz es triangular superior si todos los elementos por debajo de la diagonal principal son cero. El usuario debe ingresar n y los elementos de una matriz cuadrada $n \times n$. El programa verifica si cumple la condición.

Entrada (matriz 4×4):

```
3 5 7 1
0 2 6 9
0 0 8 4
0 0 0 5
```

Salida esperada:

```
La matriz es triangular superior.
```

Problema 10: Intercalar dos listas (2 puntos).

Escribe una función que tome dos listas de enteros y devuelva una nueva lista donde se intercalen los elementos de ambas. Si una lista es más larga, añada sus elementos restantes al final.