

Profesor: Salvador J. Vicencio-Medina

Fecha: -

Nombre: _____

ID: _____

INSTRUCCIONES GENERALES:

- **Prohibiciones:** Se prohíbe el uso de teléfonos celulares o materiales físicos (fotocopias, apuntes o similares). De no hacerlo, la evaluación se anulará sin posibilidad de apelación.
 - **Plagio:** Si durante el examen un estudiante intenta o comete plagio uso de IA, o si se le descubre intentando conversar con su(s) compañero(s), su evaluación será anulada sin posibilidad de apelación.
 - **Respuestas:** Contesta cada pregunta claramente haciendo uso del lenguaje de programación de tu preferencia.
 - **ADVERTENCIA:** Si en el examen teórico se te hace alguna pregunta de un ejercicio que contestaste en el práctico y no la respondes, dicho ejercicio queda CON PONDERACIÓN DE CERO. SIN EXCEPCIÓN.
-

1. (25 puntos) Invertir una cadena, implemente la función:

```
def invertir_cadena(cadena):
```

que reciba una cadena de texto y devuelva una nueva cadena con los caracteres en orden inverso, utilizando exclusivamente una instancia de la clase Stack proporcionada y vista en clase.

Ejemplo:

- Entrada: "estructura"
- Salida: "arutcurtse"

Restricciones:

- No utilizar slicing (`[::-1]`).
- No utilizar funciones integradas como `reversed()`.

2. (25 puntos) Evaluación de expresión postfija. Implemente la función:

```
def evaluar_postfija(expresion):
```

que reciba una expresión matemática en notación postfija (Reverse Polish Notation) separada por espacios y devuelva el resultado numérico. Utilizando exclusivamente una instancia de la clase Stack proporcionada y vista en clase.

La expresión puede contener:

- Operadores: +, -, *, /
- Números enteros positivos

Ejemplo:

- Entrada: "3 4 + 2 *"
- Salida: 14

La notación postfija (también llamada **Reverse Polish Notation**) es una forma de escribir expresiones matemáticas donde el operador se coloca **después** de los operandos.

Ejemplo comparativo:

- Notación infija tradicional: 3 + 4
- Notación postfija: 3 4 +

3. (25 puntos) Implemente una función que reciba un objeto de tipo Queue y un número entero k , y realice una rotación hacia la izquierda k veces. Utilizando exclusivamente una instancia de la clase Queue proporcionada y vista en clase.

```
def rotacion(queue,k):
```

Ejemplo:

Cola inicial: [10, 20, 30, 40, 50]
 $k = 2$

Resultado esperado: [30, 40, 50, 10, 20]

4. (25 puntos) Implemente una función que dada una lista de números enteros y un entero k , devuelva los k números más grandes utilizando una cola de prioridad con la librería heapq vista en clase.

```
def numeros_grandes(lista,k):
```

Ejemplo:

Entrada: [3, 10, 5, 20, 7], $k = 2$

Salida esperada: [20, 10]

Requisitos:

- Debe utilizar la librería heapq.
- No se permite ordenar completamente la lista.