

UNIVERSIDAD INTERAMERICANA DE PANAMÁ  
REPÚBLICA DE PANAMÁ  
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

ESTRUCTURA DE DATOS I  
EXAMEN PARCIAL 1-A (Individual)

Resuelva los siguientes problemas utilizando alguna de las estructuras de datos vistas hasta el momento (arreglos, conjuntos, registros, listas, listas enlazadas y doblemente enlazadas, pilas y colas):

Problema 1 (5 puntos):

Dado un arreglo de enteros de tamaño  $NNN$ , escribir una función que encuentre el subarreglo con la suma más grande y devuelva dicha suma.

Problema 2 (5 puntos)::

Implementa un algoritmo que ordene un arreglo de números enteros utilizando el algoritmo de Ordenamiento por Inserción y determine su complejidad temporal.

Problema 3 (5 puntos)::

Dado dos conjuntos AAA y BBB, implementa funciones para:

- Obtener la unión de ambos conjuntos.
- Obtener la intersección de ambos conjuntos.
- Obtener la diferencia  $A - B$  -  $B - A$ .

Problema 4 (5 puntos)::

Se tienen dos conjuntos de números enteros. Escribe un programa que determine si uno es subconjunto del otro.

Problema 5 (5 puntos)::

Define una estructura que almacene los datos de un estudiante (nombre, matrícula, calificación). Luego, escribe una función que reciba un arreglo de estudiantes y devuelva el nombre del estudiante con la calificación más alta.

Problema 6 (5 puntos)::

Implementa una estructura para manejar un inventario de productos con los siguientes campos: ID, nombre, cantidad y precio. Escribe un programa que permita agregar, modificar y eliminar productos del inventario.

Problema 7 (5 puntos)::

Implementa una lista simplemente enlazada en la que puedas realizar las siguientes operaciones:

- Insertar un nodo al inicio.
- Insertar un nodo al final.
- Buscar un nodo por su valor.
- Eliminar un nodo por su valor.

Problema 8 (5 puntos)::

Dada una lista doblemente enlazada de números enteros, escribe una función que la invierta en orden.

Problema 9 (5 puntos):

Implementa una pila con un arreglo dinámico y realiza las siguientes operaciones:

- push(x): Agregar un elemento.
- pop(): Eliminar el elemento en el tope.
- top(): Obtener el elemento en el tope sin eliminarlo.

Problema 10 (5 puntos):

Escribe un programa que verifique si una expresión matemática con paréntesis, corchetes y llaves está bien balanceada usando una pila.

Ejemplo:

{[{()]}) → Balanceado

{[{()]}) → No balanceado

Problema 11 (5 puntos):

Implementa una cola utilizando una estructura de datos basada en arreglos y escribe funciones para:

- enqueue(x): Agregar un elemento.
- dequeue(): Sacar un elemento.
- front(): Obtener el elemento en la parte frontal sin eliminarlo.

Problema 12 (5 puntos):

Simula un sistema de atención en una tienda donde los clientes son atendidos en orden de llegada (FIFO). Implementa la cola para manejar la fila de clientes.

Problema 13 (5 puntos):

Implementa una cola con dos pilas. Es decir, simula el comportamiento de una cola utilizando únicamente dos pilas.

Problema 14: (15 puntos):

Dada una lista enlazada con un posible ciclo, escribe un algoritmo que detecte si la lista tiene un ciclo o no (algoritmo de Floyd, "tortuga y liebre").

**Problema 15 - "El sistema de entrega de pedidos"** (20 puntos):

**Descripción:**

Una empresa de entregas tiene un sistema donde recibe pedidos de clientes y los asigna a repartidores. Los pedidos pueden ser normales o urgentes.

- Los **pedidos urgentes** deben ser atendidos primero.
- Los **pedidos normales** se atienden en el orden en que llegan.
- Cada repartidor solo puede llevar un pedido a la vez.
- Se debe permitir agregar y procesar pedidos de manera eficiente.

**Requisitos:**

- Diseñar una estructura de datos eficiente para gestionar los pedidos.

- Implementar funciones para agregar un pedido, asignar un pedido a un repartidor y mostrar los pedidos pendientes.
- El sistema debe permitir múltiples repartidores.

**Para resolverlo, deberá responder las siguientes preguntas:**

1. ¿Qué estructura de datos permite manejar prioridades?
2. ¿Cómo se podría organizar la cola para que los pedidos urgentes sean atendidos primero?
3. ¿Cómo se asignan pedidos a los repartidores de manera eficiente?