

**PRUEBA DE DIAGNÓSTICO ESTRUCTURAS DE DATOS**  
**UTFSM**

**ALUMNO:**

**1. Seleccione la opción correcta:**

**1.1. ¿Qué es una estructura de datos?**

- A. Una forma de organizar y almacenar datos \*
- B. Un lenguaje de programación
- C. Una ecuación matemática
- D. Una colección de algoritmos
- E. Una metodología de desarrollo de software

**1.2. ¿Qué es la recursividad?**

- A. Un tipo de estructura de datos.
- B. Un método para ordenar elementos.
- C. Un enfoque de programación en el que una función se llama a sí misma. \*
- D. Un algoritmo para realizar búsqueda eficiente.

**1.3. ¿Cuál es la diferencia entre una lista enlazada (linked list) y un arreglo (array)?**

- A. Una lista enlazada permite acceso aleatorio a sus elementos, mientras que un arreglo no.
- B. Una lista enlazada puede cambiar de tamaño dinámicamente, mientras que un arreglo tiene un tamaño fijo. \*
- C. Una lista enlazada almacena sus elementos contiguamente en memoria, mientras que un arreglo no.
- D. Una lista enlazada es una estructura de datos estática, mientras que un arreglo es una estructura de datos dinámica.

**1.4. ¿Cuál de las siguientes es una estructura de datos lineal?**

- A. Pila (Stack) \*
- B. Árbol binario (Binary Tree)
- C. Grafo (Graph)
- D. Montículo (Heap)
- E. Tabla Hash (Hash Table)

**1.5. ¿Qué estructura de datos sigue el principio de último en entrar, primero en salir (LIFO)?**

- A. Cola (Queue)
- B. Pila (Stack) \*

- C. Lista Enlazada (Linked List)
- D. Árbol (Tree)

- 1.6. ¿Cuál de las siguientes estructuras de datos utiliza el principio First-In-First-Out (FIFO)?
- A. Pila (Stack)
  - B. Cola (Queue) \*
  - C. Lista Enlazada (Linked List)
  - D. Árbol (Tree)
- 1.7. ¿Cuál de las siguientes estructuras de datos se utiliza típicamente para implementar una pila (stack)?
- A. Vector (std::vector)
  - B. Lista Enlazada (Linked List) \*
  - C. Árbol (Tree)
  - D. Matriz (Array)
- 1.8. ¿Cuál algoritmo de ordenamiento tiene la mejor complejidad temporal en el caso promedio?
- A. Bubble Sort
  - B. Insertion Sort
  - C. Quick Sort \*
  - D. Selection Sort
- 1.9. ¿Cuál es la complejidad temporal de insertar un elemento al final de un arreglo?
- A.  $O(1)$  \*
  - B.  $O(n)$
  - C.  $O(\log n)$
  - D.  $O(n^2)$
- 1.10. ¿Cuál es la complejidad temporal promedio de la operación de inserción en un árbol binario de búsqueda balanceado?
- A.  $O(1)$
  - B.  $O(\log n)$  \*
  - C.  $O(n)$
  - D.  $O(n^2)$
- 1.11. ¿Cuál es la complejidad temporal de buscar un elemento en un árbol de búsqueda binario balanceado?

- A.  $O(1)$
- B.  $O(\log n)$  \*
- C.  $O(n)$
- D.  $O(n^2)$

1.12. ¿Cuál es la complejidad temporal promedio de inserción en una tabla hash (hash table) con una buena función de hash?

- A.  $O(1)$  \*
- B.  $O(\log n)$
- C.  $O(n)$
- D.  $O(n^2)$

1.13. ¿Qué es un algoritmo de búsqueda binaria?

- A. Un algoritmo que busca elementos en una lista enlazada.
- B. Un algoritmo que busca elementos en un árbol binario de búsqueda.
- C. Un algoritmo que divide el conjunto de búsqueda a la mitad en cada paso. \*
- D. Un algoritmo que utiliza una función hash para buscar elementos en una tabla hash.

1.14. ¿Qué es un árbol binario de búsqueda?

- A. Un árbol en el que cada nodo tiene dos hijos.
- B. Un árbol en el que cada nodo tiene un hijo izquierdo y un hijo derecho.
- C. Un árbol en el que los elementos están ordenados y se puede realizar búsqueda eficiente. \*
- D. Un árbol en el que los elementos están organizados en niveles.

## 2. Desafíos: (factorial.cpp)

- 2.1. Implementa función en C++ para calcular el factorial de un número utilizando iteración.
- 2.2. Implementa función recursiva en C++ para calcular el factorial de un número.
- 2.3. A partir de la función anterior, escribir programa que imprima el factorial de los enteros 1 al 10.
- 2.4. ¿Cuál es la complejidad computacional de su programa
- 2.5. Proponga una implementación con complejidad lineal

## 3. Respuesta corta:

- 3.1. ¿Cuál es la diferencia entre una pila (stack) y una cola (queue)?

**Respuesta:** Una pila (stack) sigue el principio de "último en entrar, primero en salir" (LIFO), mientras que una cola (queue) sigue el principio de "primero en entrar, primero en salir" (FIFO).

3.2. Explica el concepto de recursión y proporciona un ejemplo.

**Respuesta:** La recursión es un concepto en el que una función se llama a sí misma.

3.3. ¿Cuáles son las ventajas de usar una lista enlazada (linked list) en lugar de un arreglo (array)?

**Respuesta:** Algunas ventajas de usar una lista enlazada en lugar de un arreglo son: la inserción y eliminación eficiente en cualquier posición, la capacidad de crecer dinámicamente, y la capacidad de gestionar memoria de manera más flexible.

3.4. Cuál es el propósito de la siguiente expresión: **sizeof(array)/sizeof(array[0])**

3.5. En qué momento se evalúa el operador **sizeof**

3.6. Sean las siguientes variables: suma = 10 y contador = 20, indique el valor final en suma en los siguientes casos:

3.6.1. suma += ++contador

3.6.2. suma += contador++

3.7. Considere la siguiente mala práctica de programación:  $x = ++y + y$

3.7.1. Explique la ambigüedad que se genera al momento de evaluar dicha expresión

3.7.2. Proponga 2 posibles soluciones

3.8. Explique:

```
double x; double y; const double EPSILON = 1E-14;  
if (fabs(x - y) < EPSILON)
```

#### 4. Resolución de problemas:

4.1. Dada una lista de enteros. Escribe una función para encontrar los valores máximo y mínimo en la lista. (minMax.cpp, minMaxArray.cpp)

4.2. Escribe una función en C++ para intercambiar los valores de dos enteros.

4.3. Dado un arreglo de enteros, escribe una función para encontrar el segundo elemento más grande en el arreglo.

4.4. Dado un arreglo de enteros, escribe una función para encontrar la diferencia máxima entre dos números en el arreglo.

- 4.5. Dada una cadena que contiene una combinación de letras mayúsculas y minúsculas. Escribe una función para contar el número de letras mayúsculas y minúsculas en la cadena.
- 4.6. Implementa una función en C++ que reciba una cadena de texto y devuelva la frecuencia de ocurrencia de cada carácter.
- 4.7. Implementa una función en C++ que reciba una cadena de texto y devuelva verdadero si es un palíndromo y falso si no lo es.
- 4.8. Implementa una estructura de datos de lista enlazada (linked list) en C++.
- 4.9. Escribe un fragmento de código para crear una estructura de datos de pila (stack) utilizando una lista.
- 4.10. Implementa una estructura de datos de cola (queue) utilizando dos pilas (stacks).
- 4.11. Implementa un algoritmo de búsqueda binaria.
- 4.12. Dado un árbol binario, escribe una función para determinar si es un árbol binario de búsqueda válido.
- 4.13. Escribe un fragmento de código en C++ para ordenar un arreglo de enteros utilizando el algoritmo de selección (selection sort).

## **5. Diseño:**

Diseña una clase o estructura de datos para representar un libro, con atributos como título, autor y año de publicación. Incluye métodos para obtener y establecer estos atributos.