PRÁCTICA Nº 5

**Prueba práctica de programación en C++ con std::map**

**Parte 1: Selección múltiple**

Selecciona la respuesta correcta para cada pregunta.

1. ¿Qué es **std::map** en C++? a) Una clase que define una colección de objetos de tipo **T**. b) Una estructura de datos que almacena elementos en un arreglo unidimensional. c) Una clase que define una colección de pares clave-valor.
2. ¿Cómo se insertan elementos en un **std::map**? a) Utilizando la función **insert()**. b) Utilizando la función **push\_back()**. c) Utilizando la función **emplace()**.
3. ¿Cómo se elimina un elemento de un **std::map**? a) Utilizando la función **remove()**. b) Utilizando la función **pop\_front()**. c) Utilizando la función **erase()**.
4. ¿Cómo se accede a un elemento de un **std::map**? a) Utilizando la función **get()**. b) Utilizando el operador **[]**. c) Utilizando la función **at()**.
5. ¿Qué método se utiliza para conocer el tamaño de un **std::map**? a) **count()** b) **size()** c) **length()**

**Parte 2: Ejercicios de programación**

1. Escribe un programa que cree un **std::map** de **std::string** a **int** y lo llene con los siguientes pares clave-valor:
   * "Juan" -> 20
   * "María" -> 22
   * "Pedro" -> 21
   * "Ana" -> 19
2. Utiliza una función miembro de **std::map** para mostrar el tamaño del mapa.
3. Utiliza una función miembro de **std::map** para eliminar el elemento con clave "Pedro".
4. Utiliza un bucle **for each** y la función miembro **first** y **second** de **std::pair** para mostrar el contenido del mapa.
5. Modifica el programa para que el tipo de datos del valor en el **std::map** sea una estructura que incluya la edad y el género de la persona. Modifica el programa para que pueda insertar y mostrar elementos de esta estructura.

**Parte 3: Desarrollo libre**

Desarrolla un programa utilizando **std::map** que resuelva un problema de tu elección. Este programa debe ser capaz de:

* Crear un **std::map** de algún tipo de dato.
* Insertar varios elementos en el **std::map**.
* Mostrar el tamaño del **std::map**.
* Eliminar al menos un elemento del **std::map**.
* Utilizar un bucle **for each** para mostrar los elementos del **std::map**.

Puedes elegir el problema que desees, siempre y cuando cumpla con los requisitos mencionados anteriormente.