PRÁCTICA Nº 3

Analiza en detalle este código:

1. #include <iostream>

2. #include <tuple>

3.

4. // Función que utiliza std::tuple como plantilla variádica para sumar n argumentos

5. template<typename... Ts>

6. auto sumar(Ts... args) {

7. return (args + ...);

8. }

9.

10. int main() {

11. // Llamada a la función sumar con tres argumentos de tipo int

12. int resultado1 = sumar(1, 2, 3);

13. std::cout << "El resultado de la suma es: " << resultado1 << std::endl;

14.

15. // Llamada a la función sumar con cuatro argumentos de tipo double

16. double resultado2 = sumar(1.5, 2.5, 3.5, 4.5);

17. std::cout << "El resultado de la suma es: " << resultado2 << std::endl;

18.

19. // Llamada a la función sumar con dos argumentos de tipo char

20. char resultado3 = sumar('a', 1);

21. std::cout << "El resultado de la suma es: " << resultado3 << std::endl;

22.

23. // Llamada a la función sumar con un argumento de tipo std::tuple<int, double>

24. std::tuple<int, double> tupla = std::make\_tuple(1, 2.5);

25. auto resultado4 = std::apply(sumar<int, double>, tupla);

26. std::cout << "El resultado de la suma es: " << resultado4 << std::endl;

27.

28. return 0;

29. }

30.

En este ejemplo, la función **sumar** utiliza **std::tuple** como plantilla variádica para sumar un número arbitrario de argumentos. La función toma cualquier cantidad de argumentos de cualquier tipo, y utiliza el operador de pliegue a la izquierda (**...**) para sumarlos todos.

En la función **main**, se hacen varias llamadas a la función **sumar** con diferentes tipos y números de argumentos. En la primera llamada se suman tres enteros, en la segunda se suman cuatro números de punto flotante, y en la tercera se suman un carácter y un entero.

En la cuarta llamada a **sumar**, se utiliza la función **std::apply** para aplicar la función **sumar** a los elementos de una tupla. **std::apply** toma una función como primer argumento y una tupla como segundo argumento, y aplica la función a los elementos de la tupla. En este caso, se crea una tupla de dos elementos de tipo **int** y **double**, y se llama a **std::apply** con la función **sumar<int, double>** y la tupla como argumentos. El resultado de esta llamada es un valor de tipo **double**.

Como puedes ver, **std::tuple** como plantilla variádica permite escribir funciones genéricas que pueden operar con un número variable de argumentos de diferentes tipos. Esto hace que nuestro código sea más flexible y reutilizable.

Convierte el programa anterior en otro que use una operación distinta. Por ejemplo con multiplicar es muy similar.

Puedes intentarlo también con restar aunque en este caso la sintaxis de la plantilla debe cambiar para llevar a cabo la resta en el orden correcto si los argumentos son más de dos. Puedes ver en las pistas como hacerlo.