PISTAS.

1. Pista 1: Utiliza una plantilla variádica para aceptar un número variable de argumentos de un tipo de dato específico, como se hace en el programa de ejemplo. Por ejemplo, si deseas sumar una lista de valores enteros, puedes diseñar una plantilla de la siguiente manera:

template <typename T, typename Op>

1. T average(T value)

2. {

3. return value;

4. }

5.

6. template <typename T, typename Op, typename... Args>

7. T average(T value, Args... args)

8. {

9. return Op()(value, average<T, Op>(args...));

10. }

11.

1. Pista 2: Utiliza una plantilla de operación para permitir que el usuario especifique el tipo de operación que se realizará en los argumentos de entrada. Puedes hacer esto proporcionando un parámetro de plantilla adicional que acepte un tipo de operación. Por ejemplo, si deseas calcular la media de una lista de valores de punto flotante, puedes utilizar la siguiente plantilla de operación:

1. template <typename T, typename Op>

2. T average(T value)

3. {

4. return value;

5. }

6.

7. template <typename T, typename Op, typename... Args>

8. T average(T value, Args... args)

9. {

10. return Op()(value, average<T, Op>(args...));

11. }

12.

1. Pista 3: Utiliza las funciones **std::cout** y **typeid** para depurar tu código. Por ejemplo, si tienes problemas para determinar si los argumentos de entrada se están pasando correctamente, puedes imprimir el tipo de datos de cada argumento en la función **average** de la siguiente manera:

1. template <typename T, typename Op, typename... Args>

2. T average(T value, Args... args)

3. {

4. std::cout << typeid(value).name() << '\n';

5. std::cout << typeid(Op).name() << '\n';

6.

7. return Op()(value, average<T, Op>(args...));

8. }

9.

Esto imprimirá el nombre del tipo de datos y el nombre del tipo de operación en la consola para ayudarte a identificar cualquier problema.

1. Pista 4: Verifica que estás proporcionando el tipo de operación correcto en tu plantilla. Por ejemplo, si deseas calcular la media de una lista de valores de punto flotante utilizando la operación **std::divides**, debes proporcionarla como un parámetro de plantilla al llamar a la función **average** de la siguiente manera:

1. double result = average<double, std::divides<double>>(1.0, 2.0, 3.0, 4.0, 5.0);

1. Pista 5: Si tienes dificultades para comprender las plantillas variádicas en C++, repasa los conceptos fundamentales y trabaja sobre las plantillas ya creadas en el curso.