PRÁCTICA Nº 4

Ejercicio: Crea una serie de plantillas de función que realicen operaciones matemáticas simples, utilizando diferentes configuraciones de tipos de parámetros y tipos de retorno. Cada plantilla de función debe tomar dos parámetros y devolver un valor calculado. Las operaciones matemáticas que se deben implementar son:

1. Suma: toma dos números y devuelve la suma.
2. Resta: toma dos números y devuelve la diferencia.
3. Multiplicación: toma dos números y devuelve el producto.
4. División: toma dos números y devuelve el cociente.
5. Potencia: toma un número base y un exponente y devuelve la base elevada al exponente.

Crea las plantillas de función utilizando las siguientes variantes:

1. Una plantilla de función que use tipos de parámetros definidos dentro de los paréntesis angulares y que devuelva un tipo de retorno definido dentro de los paréntesis angulares.
2. Una plantilla de función que use tipos de parámetros definidos dentro de los paréntesis angulares y que devuelva el tipo de retorno auto.
3. Una plantilla de función que use tipos de parámetros auto y que devuelva un tipo de retorno definido dentro de los paréntesis angulares.
4. Una plantilla de función que use tipos de parámetros auto y que devuelva el tipo de retorno auto.

Utiliza los ejemplos siguientes como base para crear cada plantilla de función:

// Plantilla de función que usa tipos definidos dentro de los paréntesis angulares y que devuelve un tipo definido dentro de los paréntesis angulares.

template<typename T>

T suma(T a, T b) {

return a + b;

}

// Plantilla de función que usa tipos definidos dentro de los paréntesis angulares y que devuelve el tipo de retorno auto.

template<typename T>

auto resta(T a, T b) {

return a - b;

}

// Plantilla de función que usa tipos de parámetros auto y que devuelve un tipo definido dentro de los paréntesis angulares.

template<typename T>

T multiplicacion(auto a, T b) {

return a \* b;

}

// Plantilla de función que usa tipos de parámetros auto y que devuelve el tipo de retorno auto.

template<typename T, typename U>

auto division(T a, U b) {

return a / b;

}