# TASK 1:

#include<iostream>

using namespace std;

template<class A>

A add(A a, A b)

{

return a + b;

}

template<class M>

M Mul(M a, M b)

{

return a \* b;

}

int main()

{

cout << "Addition in int: " << add<int>(3, 4) << endl;

cout << "Addition in float: " << add<float>(3.1, 4.2) << endl;

cout << "Addition in double: " << add<double>(3.222333, 4.333222) << endl << endl;

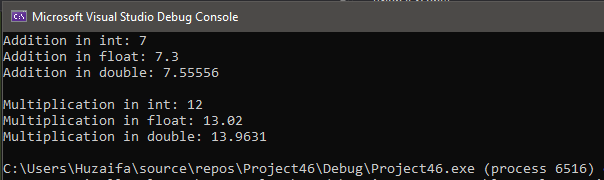
cout << "Multiplication in int: " << Mul<int>(3, 4) << endl;

cout << "Multiplication in float: " << Mul<float >(3.1, 4.2) << endl;

cout << "Multiplication in double: " << Mul<double>(3.222333, 4.333222) << endl;

# }

# OUTPUT:



# TASK 2:

#include<iostream>

using namespace std;

template<class A>

A avg(A arr[], A b)

{

A a = 0;

for (int i = 0; i < b; i++)

{

a += arr[i];

}

return a;

}

int main()

{

int arr[3];

long arr1[3];

double arr2[3];

char arr3[3];

cout << "Enter array for int: " << endl;

for (int i = 0; i < 3; i++)

{

cout << "Array " << i << ":";

cin >> arr[i];

}

cout << endl;

cout << "Enter array for long: " << endl;

for (int i = 0; i < 3; i++)

{

cout << "Array " << i << ":";

cin >> arr1[i];

}

cout << endl;

cout << "Enter array for double: " << endl;

for (int i = 0; i < 3; i++)

{

cout << "Array " << i << ":";

cin >> arr2[i];

}

cout << endl;

cout << "Enter array for char: " << endl;

for (int i = 0; i < 3; i++)

{

cout << "Array " << i <<":";

cin >> arr3[i];

}

cout << endl;

cout << "Addition in int: " << avg<int>(arr, 3) << endl;

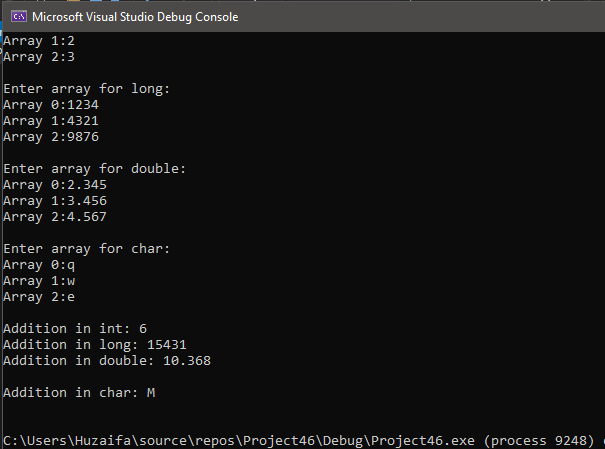
cout << "Addition in long: " << avg<long>(arr1, 3) << endl;

cout << "Addition in double: " << avg<double>(arr2, 3) << endl << endl;

cout << "Addition in char: " << avg<char>(arr3, 3) << endl << endl;

}

# OUTPUT:



# TASK 3:

#include<iostream>

using namespace std;

template<class S>

void Swap(S& a, S& b)

{

S temp;

temp = a;

a = b;

b = temp;

}

int main()

{

int a = 4, b = 9;

float a1 = 2.2, b1 = 3.3;

double a2 = 1234, b2 = 9876;

char a3 = 'q', b3 = 'w';

string a4 = "qwerty", b4 = "asdfgh";

cout << "Without Swapping in int: " << a << " " << b << endl;

cout << "Without Swapping in float: " << a1 << " " << b1 << endl;

cout << "Without Swapping in double: " << a2 << " " << b2 << endl;

cout << "Without Swapping in char: " << a3 << " " << b3 << endl;

cout << "Without Swapping in string: " << a4 << " " << b4 << endl << endl;

Swap(a, b);

Swap(a1, b1);

Swap(a2, b2);

Swap(a3, b3);

Swap(a4, b4);

cout << "Swapping in int: " << a << " " << b << endl;

cout << "Swapping in float: " << a1 << " " << b1 << endl;

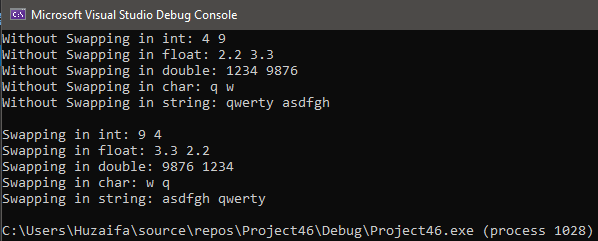
cout << "Swapping in double: " << a2 << " " << b2 << endl;

cout << "Swapping in char: " << a3 << " " << b3 << endl;

cout << "Swapping in string: " << a4 << " " << b4 << endl;

}

# OUTPUT:



# TASK 4:

#include<iostream>

using namespace std;

class Triangle

{

public:

Triangle (){}

template<typename S>

S Area(S length, S width)

{

S temp = 0;

temp = (length \* width) / 2;

return temp;

}

template<typename S>

S Peri(S length, S width)

{

S temp = 0;

temp = length + width;

return temp;

}

};

int main()

{

Triangle obj;

//Area Functions

cout << "Area(int,int) : " << obj.Area<int>(5, 9) << endl;

cout << "Area(float,float) : " << obj.Area<float>(12.5, 9.7) << endl;

cout << "Area(double,double) : " << obj.Area<double>(14.001, 12.112) << endl;

cout << "Area(int,float) : " << obj.Area<int>(12, 13.53) << endl;

cout << "Area(float, double) : " << obj.Area<float>(13.53, 12.00091) << endl;

cout << "Area(double, int) : " << obj.Area<double>(12.00091, 9) << endl << endl;

//Perimeter Functions

cout << "Peri(int,int) : " << obj.Peri<int>(5, 9) << endl;

cout << "Peri(float,float) : " << obj.Peri<float>(12.5, 9.7) << endl;

cout << "Peri(double,double) : " << obj.Peri<double>(14.001,12.112) << endl;

cout << "Peri(int,float) : " << obj.Peri<int>(12, 13.53) << endl;

cout << "Peri(float, double) : " << obj.Peri<float>(13.53, 12.00091) << endl;

cout << "Peri(double, int) : " << obj.Peri<double>(12.00091, 9) << endl;

}

# OUTPUT:

