**Assignment#09-PF-Thoery**

**Attempt all questions. Submission deadline is 10-July-2020.**

**Question #01**

Using the concept of **Function overloading** make the 4 UDF of same name *solve* which having the following functionality:

1) Solve (First) {Doesn’t return any value but pass the integer value to the UDF and UDF draws its table}.

2) Solve (Second) {Doesn’t return any value but pass the three float values to the UDF and UDF computes the sum of square of these values}.

3) Solve (Third) {Return float value but pass the four integer values to the UDF and UDF computes the average of these values and returns average}.

4) Solve the part (2) using **function template** and same function name but sending 3 integer values (for first UDF execution) and 3 float values (for second UDF execution).

**SOURCE CODE**

#include<iostream>

#include<string>

using namespace std;

void solve(int a){

for(int i=0; i<=12; i++)

{

cout<<a<<" \* "<<i<<" = "<<a\*i<<endl;

}

}

void solve(float a, float b,float c){

float sum;

sum = (a\*a)+(b\*b)+(c\*c);

cout<<"The sum of square of values is: "<<sum<<endl;

}

float solve(int a, int b, int c, int d){

float avg,sum;

sum = a+b+c+d;

avg = sum /4;

return avg;

}

template <typename t>

t solve(t a, t b,t c){

float sum;

sum = (a\*a)+(b\*b)+(c\*c);

cout<<"The sum of square of values is: "<<sum<<endl;

}

int main() {

float a,b,c,d,avg;

cout<<"Calling first UDF: "<<endl;

cout<<"Enter an integer value: "<<endl;

cin>>a;

solve(a);

cout<<endl<<"Calling second UDF: "<<endl;

cout<<"Enter 3 float values to calculate sum of square of values: "<<endl;

cin>>a>>b>>c;

solve(a,b,c);

cout<<endl<<"Calling third UDF: "<<endl;

cout<<"Enter 4 integer values for average: "<<endl;

cin>>a>>b>>c>>d;

avg = solve(a,b,c,d);

cout<<"The average of the values is: "<<avg<<endl;

cout<<endl<<"Calling fourth UDF: "<<endl;

cout<<"Enter 3 integer values to calculate sum of square of values: "<<endl;

cin>>a>>b>>c;

solve<int>(a,b,c);

cout<<endl<<"Calling fourth UDF: "<<endl;

cout<<"Enter 3 values in float to calculate sum of square of values: "<<endl;

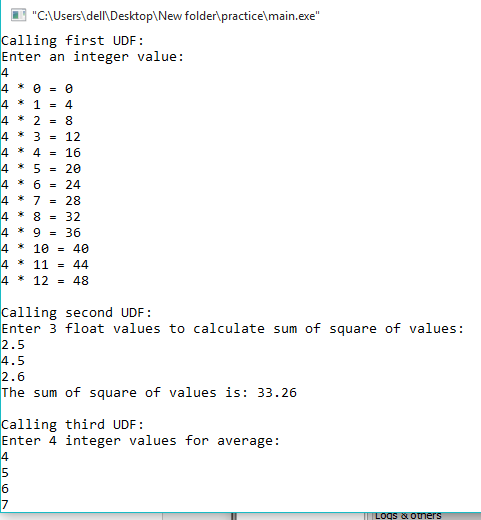
cin>>a>>b>>c;

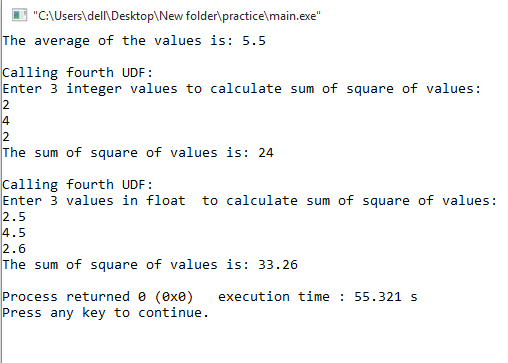
solve<float>(a,b,c);

return 0;

}

**OUTPUT**





**Question #02**

Write a program that inputs two characters from user into normal character variable and change their cases. If both characters are small alphabets then convert them into capital alphabets, if both are capital alphabets then convert them in small alphabets, if anyone of them is small and other is capital then change their cases using the pointer variables.

Output

*Enter an alphabet: h*

*Enter an alphabet: Y*

*After changing case output is H and y*

**SOURCE CODE**

#include<iostream>

using namespace std;

void convertOpposite(string &str)

{

int ln = str.length();

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

{

if (str[i]>='a' && str[i]<='z')

str[i] = str[i] - 32;

else if(str[i]>='A' && str[i]<='Z')

str[i] = str[i] + 32;

}

}

int main()

{

string str,str1;

cout<<"Enter an alphabet: ";

cin>>str;

cout<<"Enter an alphabet: ";

cin>>str1;

convertOpposite(str);

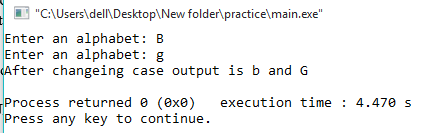
convertOpposite(str1);

cout <<"After changeing case output is "<< str <<" and "<< str1 <<endl;

return 0;

}

**OUTPUT**



**Question #03**

Write a program that inputs two integer values from user using pointer notation. Pass these values to a function using pointers. The function then returns a pointer stored ***first – second*** if first value is greater than second and returns ***first + second*** otherwise. Display the result in main.

Output

*Enter first number: 13*

*Enter second number: 35*

*Final result is: 48*

**SOURCE CODE**

#include<iostream>

using namespace std;

int func(int\*, int\*);

int func(int\* first, int\* second)

{

if (\*first>\*second)

{

return (\*first - \*second);

}

else

{

return (\*first + \*second);

}

}

int main()

{

int str,str1,output;

cout<<"Enter First number: ";

cin>>str;

cout<<"Enter Second number: ";

cin>>str1;

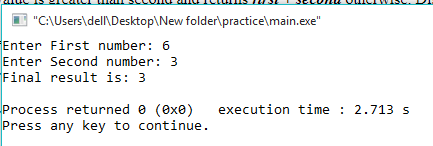
output = func(&str , &str1);

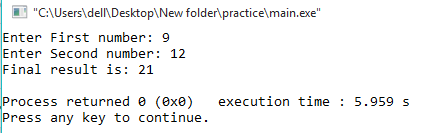
cout <<"Final result is: "<< output <<endl;

return 0;

}

**OUTPUT**

****

****