**FUNCTIONS**

**LAB # 5**



**Spring 2019**

**CSE102L Computer Programming Lab**

Submitted by: **SHAH RAZA**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Madiha Sher**

March 27, 2019

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

## Objectives:

To understand function programming, its types and function-call.

**TASK #1:**

Write a program that takes marks as input and displays the grade using function.

**Code:**

#include <iostream>

using namespace std;

void Grade(int a) //Defining Grade Function

{

switch(a)

{

case 91 ...100:

cout<<"Grade= A";

break;

case 81 ...90:

cout<<"Grade= B";

break;

case 71 ...80:

cout<<"Grade= C";

break;

case 61 ...70:

cout<<"Grade= D";

break;

case 0 ...60:

cout<<"Grade= F";

break;

default:

cout<<"Invalid Entry";

}

}

int main()

{

int marks; //Variable declaration

cout<<"Enter Marks: "; //Display message

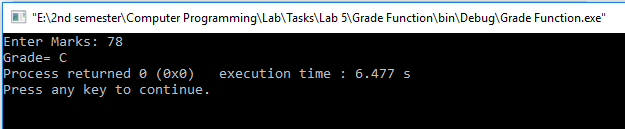
cin>>marks;//Input marks

Grade(marks); //Calling the Grade function

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #2:**

Write a function minmax() that takes four integers as input and display the minimum and maximum number.

**Code:**

#include <iostream>

using namespace std;

void min\_max(int a,int b,int c,int d) //Definition of min\_max function

{

if(a>b && a>c && a>d)

cout<<"Max: "<<a<<endl;

else if(b>a && b>c && b>d)

cout<<"Max: "<<b<<endl;

else if(c>a && c>b && c>d)

cout<<"Max: "<<c<<endl;

else

cout<<"Max: "<<d<<endl;

if(a<b && a<c && a<d)

cout<<"Min: "<<a<<endl;

else if(b<a && b<c && b<d)

cout<<"Min: "<<b<<endl;

else if(c<a && c<b && c<d)

cout<<"Min: "<<c<<endl;

else

cout<<"Min: "<<d<<endl;

}

int main()

{

int a,b,c,d; //Variable declaration

cout << "Enter a:"; //Display message

cin>>a; //Input a

cout << "Enter b:"; //Display message

cin>>b; //Input a

cout << "Enter c:"; //Display message

cin>>c; //Input a

cout << "Enter d:"; //Display message

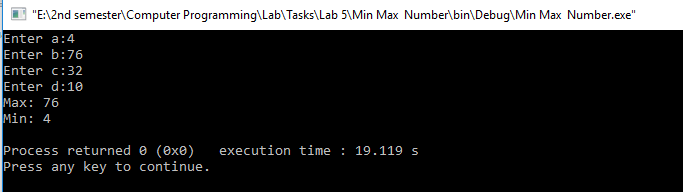
cin>>d; //Input a

min\_max(a,b,c,d); //Calling min\_max function

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #3:**

Write a program that prints all prime numbers between 2 and 1000 inclusive. Your program should have a function named ‘prime’ which accepts an integer and return a Boolean (a true if the number is prime and false otherwise). Display the numbers in main().

**Code:**

#include <iostream>

using namespace std;

bool Prime(int a) //Definition of Prime function

{

int counter=0;

for(int i=2;i<=a;i++)

{

if(a%i==0)

{

counter++;

}

}

if(counter>1)

{

return 0;

}

return 1;

}

int main()

{

cout<<"Prime Number between 2 to 1000:\n"; //Display message

for(int i=2;i<=1000;i++ ) //Loop repetition from 2 to 1000

{

int test=Prime(i); //Calling the Prime function

if(test==1)

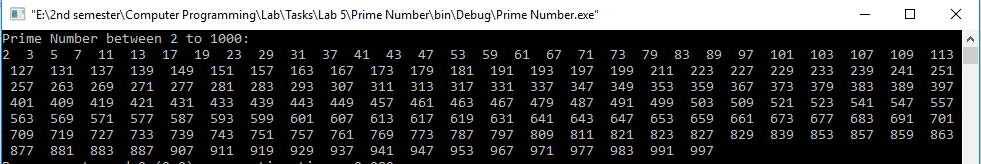
cout<<i<<" "; //Display message

}

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #4:**

Given an integer number; you have to find the total number of minimum bit(s) which can be used to store given integer number. Implement the program using function. Function will take the integer as input and return the number of bits required.

**Code:**

#include <iostream>

using namespace std;

int bits(int a) //Definition of bits function

{

int b=0;

while(a!=0)

{

b++;

a=a>>1;

}

return b;

}

int main()

{

int num,out; //Variable declaration

cout << "Enter a number: "; //Display message

cin>>num; //Input num

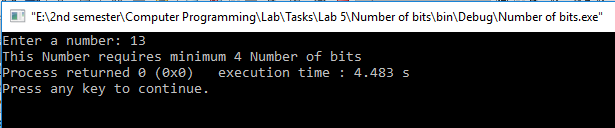
out=bits(num); //Calling bits function

cout<<"This Number requires minimum "<<out<<" Number of bits"; //Display number of bits

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #5:**

Write a C++ function to count number of 1’s in an integer number.

**Code:**

#include <iostream>

using namespace std;

int number\_of\_1s (int a) //Definition of number\_of\_1s function

{

int counter=0,remainder;

while(a!=0)

{

remainder=a%10;

if(remainder==1)

counter++;

a=a/10;

}

return counter;

}

int main()

{

int num,out; //Variable declaration

cout << "Enter a number: "; //Display message

cin>>num; //Input num

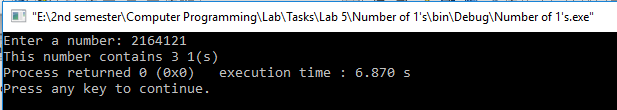
out= number\_of\_1s(num); //Calling number\_of\_1s function

cout<<"This number contains "<<out<<" 1(s)"; //Display number of 1(s)

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #6:**

Write a program to check the status of a particular bit of a number using function. Function will take an integer and bit number as input and return a Boolean value. Return true if bit was high, return false if the bit was low.

**Code:**

#include <iostream>

using namespace std;

bool bit(int a, int b) //Definition of bit function

{

a=a>>(b-1);

bool Bit=a&1;

return Bit;

}

int main()

{

int num,bit\_number; //Variable declaration

cout << "Enter a number:"; //Display message

cin>>num; //Input num

cout<<"Which bit of "<<num<<" do you want to check? "; //Display message

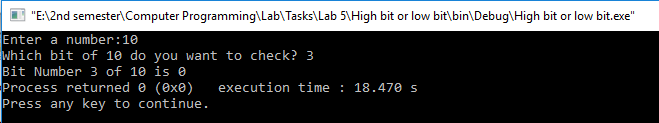
cin>>bit\_number; //Input bit\_number

cout<<"Bit Number "<<bit\_number<<" of "<<num<<" is "<<bit(num,bit\_number);

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #7:**

Write a program to find the roots of a quadratic equation of type a.x2+b.x+c where a is not equal to zero.

**Code:**

#include <iostream>

#include <math.h>

using namespace std;

float roots(float a, float b, float c, float &r1, float &r2) //Definition of roots function

{

float d=(b\*b)-(4\*a\*c);

if (d>0)

{

r1=(-b+sqrt(d))/(2\*a);

r2 = -(b + sqrt(d)) / (2\*a);

}

else if(d==0)

{

r1 = -(b) / (2\*a);

r2=r1;

}

else

{

r1= -b /(2 \* a);

r2=sqrt(-d) / (2\*a);

}

}

int main()

{

float a,b,c,r1,r2; //Variable declaration

cout << "Enter a: " ; //Display message

cin>>a; //Input a

cout << "Enter b: " ; //Display message

cin>>b; //Input b

cout << "Enter c: " ; //Display message

cin>>c; //Input c

roots(a,b,c,r1,r2);

cout<<"Roots:\n"; //Display message

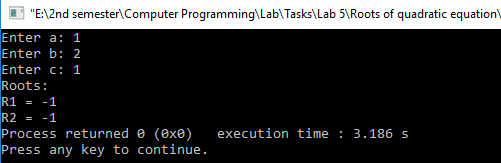
cout<<"R1 = "<<r1; //Display root1

cout<<"\nR2 = "<<r2; //Display root2

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #8:**

Write Program to compute Sin(x) and Cos(x) using Taylor series approximation given by

Sin(x) = x - (x3/3!) + (x5/5!) - (x7/7!) + …….

Cos(x) = 1 - (x2/2!) + (x4/4!) - (x6/6!) + …….

Compare the result with the built- in Library function and print both the results.

**Code:**

#include <iostream>

#include <math.h>

#define PI 3.14159265

int factorial(int a) //Definition of factorial function

{

int fact=1;

for(int i=1;i<=a;i++)

{

fact=fact\*i;

}

return fact;

}

float mysin(float deg) //Definition of mysin function

{

float sum= 0;

int check=0,sign;

for(int i=1;i<=15;i++)

{

if(i%2!=0)

{

check++;

if(check%2==0)

sign=-1;

else

sign=1;

float nominator=pow(deg,i);

int denominator= factorial(i);

float expression=(nominator\*sign)/denominator;

sum=sum+expression;

}

}

return sum;

}

float mycos(float deg) //Definition of mycos function

{

float sum=1;

int check=0,sign;

for(int i=1;i<=15;i++)

{

if(i%2==0)

{

check++;

if(check%2==0)

sign=1;

else

sign=-1;

float nominator= pow(deg,i);

int denominator= factorial(i);

float expression=(nominator/denominator)\*sign;

sum=sum+expression;

}

}

return sum;

}

using namespace std;

int main()

{

double param,result,result2,result3,result4; //Variable declaration

cout<<"Enter The Angle in Degrees: "; //Display message

cin>>param; //Input param

result= sin(param\*PI/180); //Built-in sin function

result2= mysin(param\*PI/180); //mysin function

result3= cos(param\*PI/180); //Built-in cos function

result4= mycos(param\*PI/180); //mycos function

cout<< "The sine of " << param <<" degree using built-in function is " << result<<endl;

cout<<"The sine of "<<param<<" degree using user defined function is " << result2<<endl;

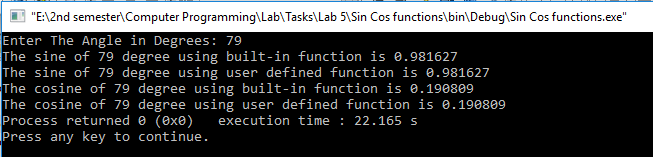
cout<< "The cosine of " << param <<" degree using built-in function is " << result3<<endl;

cout<<"The cosine of "<<param<<" degree using user defined function is " << result4;

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #9:**

A “Perfect” number is a positive whole number that is the sum of its proper divisors (including 1 and excluding the number itself). For example, the proper divisors of 6 are 1, 2, 3 and 1 + 2 + 3 = 6. So, 6 is a perfect number. Similarly, 28 is also a perfect number.

Write a program that displays first 4 perfect numbers. The program should be composed of at least two functions additional to main function, one that accepts a number and returns a Boolean true if the number is perfect and false otherwise. The other function should display all the proper divisors of the perfect number.

**Code:**

#include <iostream>

using namespace std;

void divisors(int a) //Definition of divisors function

{

for(int i=1;i<a;i++)

{

if(a%i==0)

{

cout<<i<<" ";

}

}

}

bool Perfect(int a) //Definition of Perfect function

{

int sum=0;

for(int i=1;i<a;i++)

{

if (a%i==0)

{

sum=sum+i;

}

}

if (sum==a)

{

return 1;

}

else

return 0;

}

int main()

{

int counter=0;

for(int i=1;i>=0;i++)

{

int test=Perfect(i);

if (test==1)

{

counter++;

cout<<i<<" is a Perfect Number\n";

cout<<"Its Divisors are: ";

divisors(i);

cout<<endl;

}

if(counter==4)

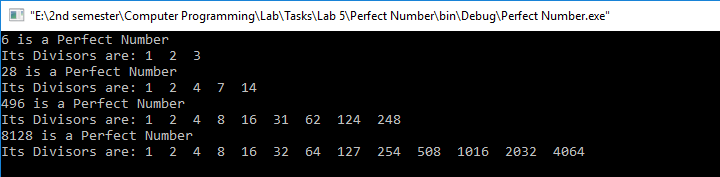
return 0;

}

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #10:**

Write a program that displays the factorial of first N integers. Use function to find factorial.

**Code:**

#include <iostream>

using namespace std;

int factorial(int a) //Definition of factorial function

{

int fact=1;

for(int i=1;i<=a;i++)

{

fact=fact\*i;

}

return fact;

}

int main()

{

int N; //Variable declaration

cout << "Enter N:"; //Display message

cin>>N; //Input N

for(int i=1;i<=N;i++)

{

int out=factorial(i); //Calling factorial function

cout<<"Factorial of "<<i<<" is "<<out<<endl; //Display factorial of i

}

return 0;

}

**Output (Compilation, Debugging and Testing):**

