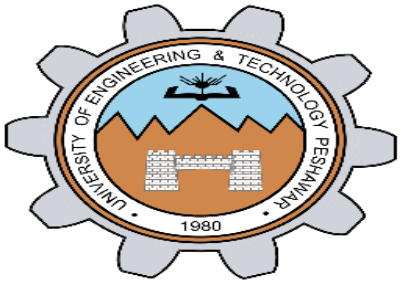
LAB REPORT NO 7



Spring 2020

CSE102L Computer Programming Lab

Submitted by: **Muhammad Ali**

Registration No. : **19PWCSE1801**

Class Section: A

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

Submitted to:

Engr. Abdullah Hamid

(July 16, 2020)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

1. Print 1 to 100 in C++ using recursion.

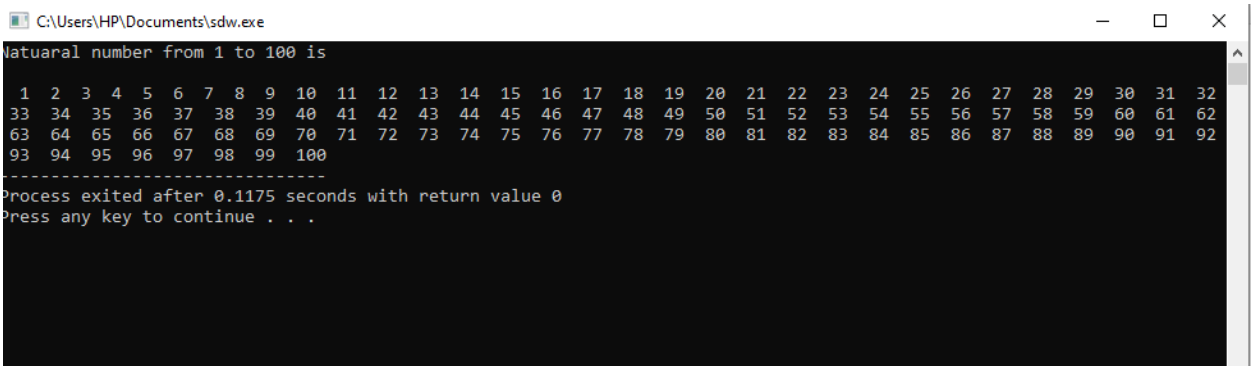
```
#include<iostream>
using namespace std;
 int nat(int n);
 int main(){
   int n=100;
   cout<<"Natuaral number from 1 to 100 is\n"<<endl;
     for(int i=1; i<=n; i++){
       cout<<"  "<< nat (i);


     }



    return 0;
 }
int nat(int n)
 {
      return (n);


 }
```



2. Write a C++ program where you take two values from user if the user enter one or two of

the values zero instead of passing the zero values to the function let the function calculate

default values if user enters values other than zero pass them to function and calculate

their sum.

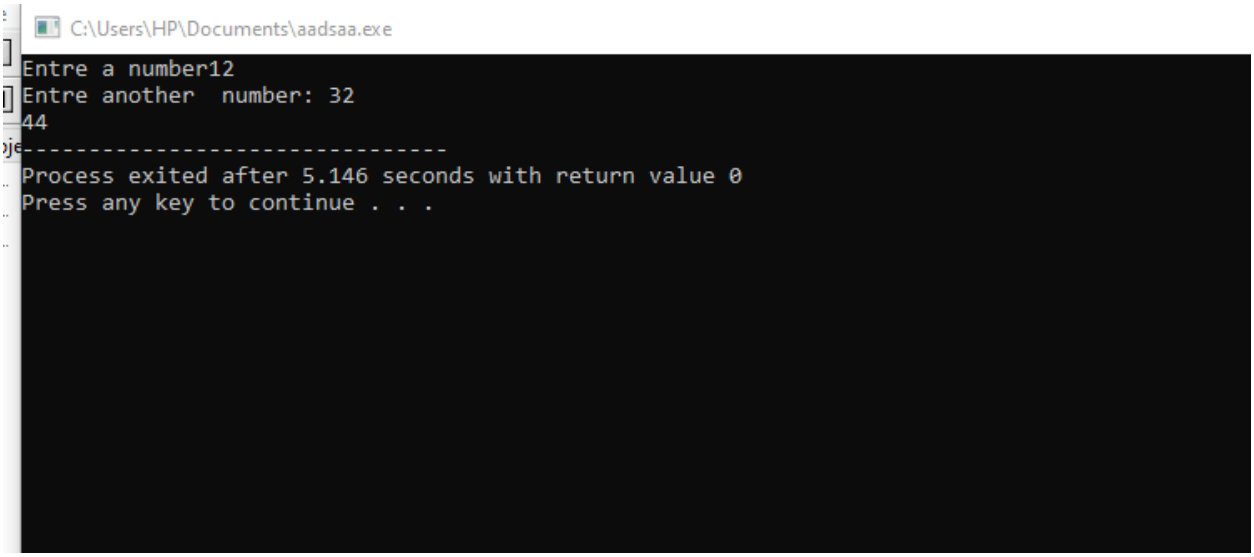
```
#include<iostream>
using namespace std;
int sum(int x = 3, int y = 6);
main() {
        int a, b;
        cout<<"Entre a number";
        cin>>a;
```

```
	cout<<"Entre another  number: ";
	cin>>b;

	if((a==0)&&(b==0)) {
		cout<<sum();

	}else if((a==0) || (b==0)) {
		if(a==0) cout<<sum(b);
		else cout<<sum(a);
	}else {
		cout<<sum(a,b);
	}


}
int sum(int x, int y) {
	return x + y;
}
```



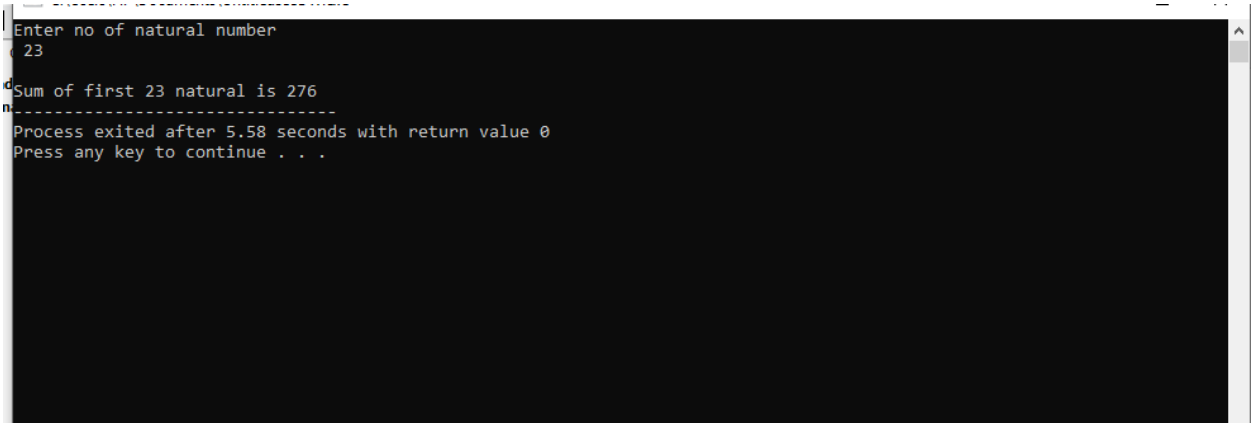
3. Write a function to find Sum of N natural numbers using Recursion.

```
#include<iostream>
using namespace std;
 int add(int n);
 int main(){
        int n;
        cout<<"Enter no of natural number\n ";
         cin>>n;
          cout<<"\nSum of first "<<n<<" natural is ";
           cout<<add(n);



 return 0;
 }
int add(int n)
{
      if (n!=0)
      return n +add(n-1);
return 0;

 }
```



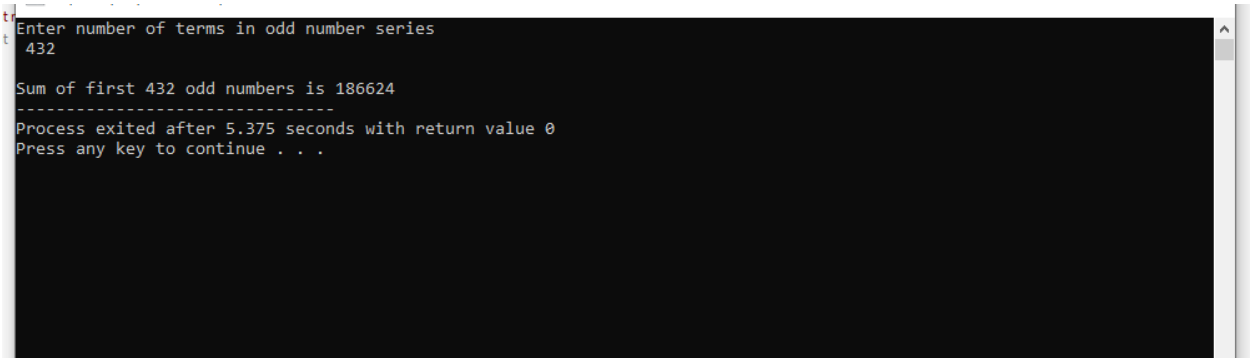
4. Calculate the sum of odd natural numbers 1+3+5+7+……. + n using Recursion.

Take n as input from user.

```
#include<iostream>
using namespace std;
 int addodd(int n);
 int main(){
        int n;
        cout<<"Enter number of terms in odd number series\n ";
         cin>>n;
             cout<<"\nSum of first "<<n<<" odd numbers is ";
       cout<<addodd(n);
 return 0;
 }
int addodd(int n)
 {
    if (n==1)
```

```
    return (1);
  else
return (addodd(n-1)+(2*n-1));

}
```



5. Overload three functions with name grade() one function should accept marks(int) and

output the percentage and the other function should accept the percentage(float) and

display the grade based on the percentage from previous function finally the third

function also named grade should make a call to these other two functions

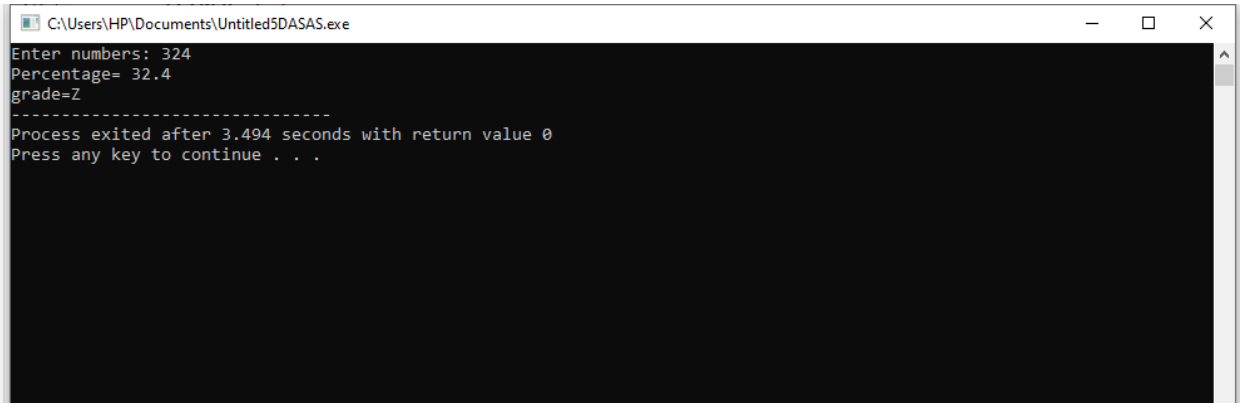
```
#include<iostream>
```

```
using namespace std;
int marks;
void grade();
float grade(int a);
char grade(float b);
main() {
	cout<<"Enter numbers: ";
	cin>>marks;
	grade();
}



float grade(int a) {
	float p = (float) a/1000;
	p*=100;
	return p;
}
char grade(float b) {
	if(b > 90) {
		return 'A';
```

```
        }else if(b > 80) {
                return 'B';
        }else if(b > 70) {
                return 'C';
        }else if(b > 60) {
                return 'D';
        }else if(b > 50) {
                return 'E';
        }else if(b > 40) {
                return 'G';
        }else if(b > 30) {
                return 'Z';
        }else {
                return 'F';
        }
}
void grade() {
        float per = grade(marks);
        char gr = grade(per);
        cout<<"Percentage= "<<per<<"\ngrade="<<gr;
}
```



6. Write a C++ Program to Find Factorial of a Number Using Recursion.

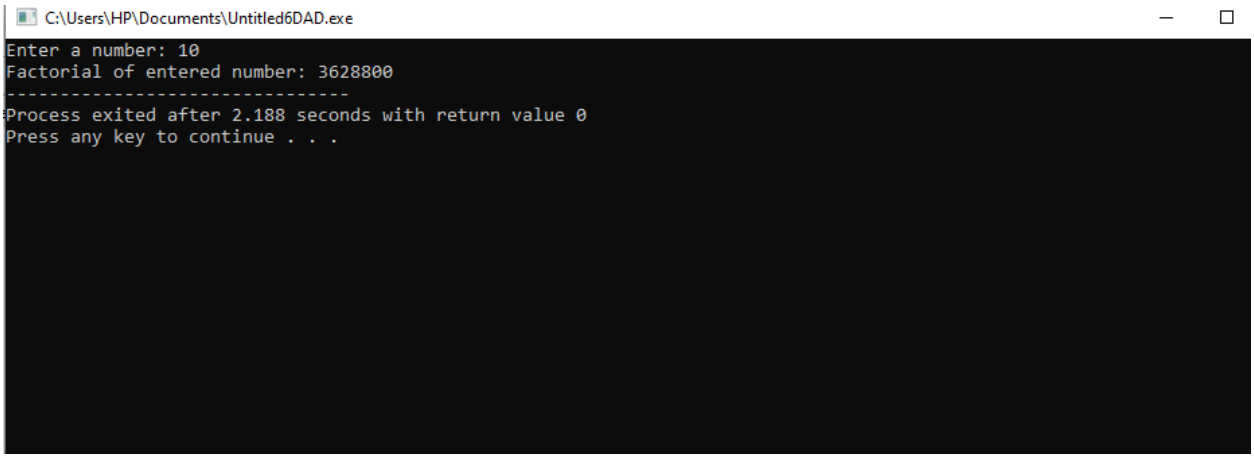
```
#include <iostream>

using namespace std;

//Factorial function

int f(int n){
```

```
  /* This is called the base condition, it is
   * very important to specify the base condition
   * in recursion, otherwise your program will throw
   * stack overflow error.
   */
  if (n <= 1)
      return 1;
  else
     return n*f(n-1);
}
int main(){
  int num;
  cout<<"Enter a number: ";
  cin>>num;
  cout<<"Factorial of entered number: "<<f(num);
  return 0;
}
```



7. C++ Program to Find L.C.M Using Recursion.

```
#include <iostream>
using namespace std;
int lcm(int x, int y) ;
int main(){

int x,y;
cout<<"Enter two numbers\n";
cin>>x>>y;
cout<<"LCM of "<<x<<" and "<<y<<" is equal to "<<lcm(x,y)<<endl;
  return 0;
}
int lcm(int x, int y){
   static int m=0;
   m=m+y;
   if((m%x==0)&& (m%y==0))
   {
      return m;
   }
 else{
   return lcm(x,y);
 }
}
```

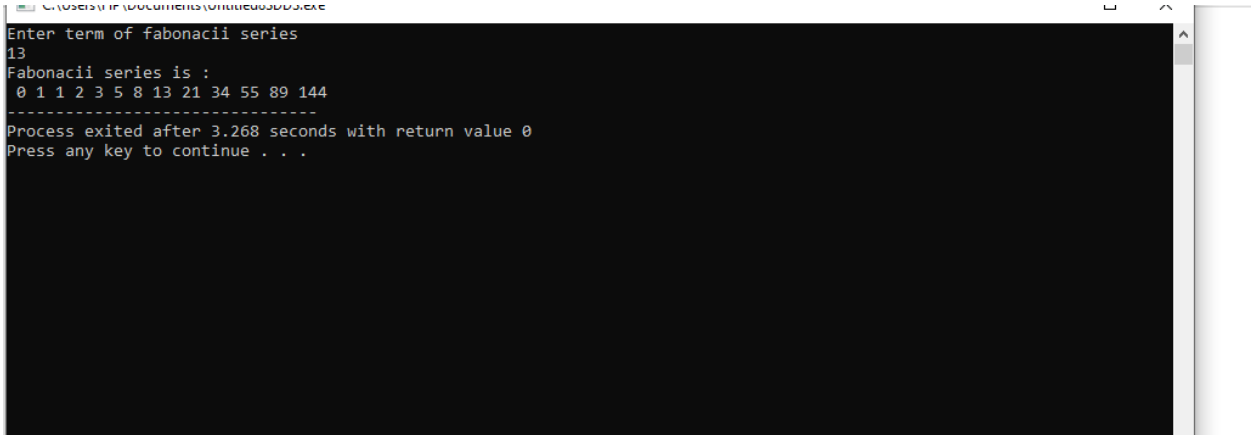
8. C++ program to print Fibonacci series using recursion.

```
#include <iostream>
using namespace std;
int fabo(int x) {
  if((x==1)||(x==0)) {
    return(x);
}else{
  return (fabo(x-1)+fabo(x-2));
 }
}
int main(){

int x;
cout<<"Enter term of fabonacii series\n";
cin>>x;
```

```
cout<<"Fabonacii series is :\n";
for(int i=0; i<x; i++){
  cout<<" "<< fabo(i);
}

  return 0;
}
```



9. C++ program to calculate power of a number using recursion.

```
#include<iostream>
using namespace std;
 int pow(int n ,int x);
 int main(){
        int n,x;
        cout<<"Enter a number\n ";
         cin>>n;
cout<<"Enter a number\n ";
```

```
cin>>x;
cout<<n<<"^"<<x<<" = ";
      cout<<pow(n,x);
 return 0;
 }
int pow(int n ,int x)
 {
    if (x==0)
return (1);
    else
       return n*pow(n,x-1);
 }
```