	Department of Computer Science and Engineering Faculty of Engineering, South Eastern University of Sri Lanka		
	Subject	**CS53003: Data Structure and Algorithms**	
	Batch	**E18**	**Semester**	**5**

Lab no and title : Lab 02: Iteration and Recursion
Name : G.W.P.R.R. Wijesinghe
Reg No : SEU-IS-18-EG-013
Submission Date : 25-12-2022

---

1. Write a recursive function that computes the sum of all numbers from 1 to n, where n is given as parameter. Test the program for input n = 100;

Start here | EX01.cpp

```cpp
#include<iostream>

using namespace std;

int CalculateSum(int m){
    if (m!=0){
        return m+CalculateSum(m-1);
    }
}

int main(){
    int n;
    cout<<"Enter the number of elements :";
    cin>>n;
    cout<<"Total is "<<CalculateSum(n);
}
```

```
"E:\Campus Semseters\5th Semester\CS 53003 Data Structure and Algorithms\Lab\02\Final\EX0...
Enter the number of elements :100
Total is 5050
Process returned 0 (0x0)   execution time : 1.884 s
Press any key to continue.
```

2. Write C++ programs that use both recursive and non-recursive functions to find the factorial of a given integer.

Non-recursive

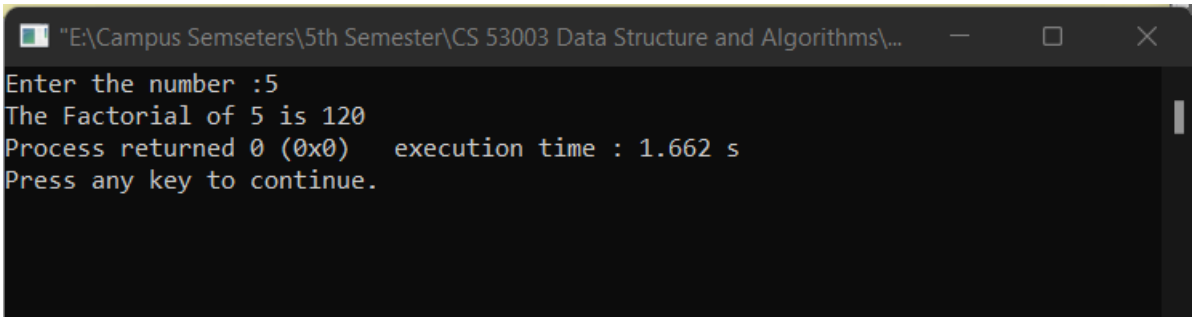
Start here | EX01.cpp | EX02part01.cpp

```cpp
#include<iostream>

using namespace std;

int main(){
    int n;
    cout<<"Enter the number :";
    cin>>n;
    int Factorial = 1;
    for (int i=1;i<=n;i++){
        Factorial = Factorial*i;
    }
    cout<<"The Factorial of "<<n<<" is "<<Factorial;

}
```



Recursive

Start here | EX01.cpp | EX02part01.cpp | EX02part2.cpp

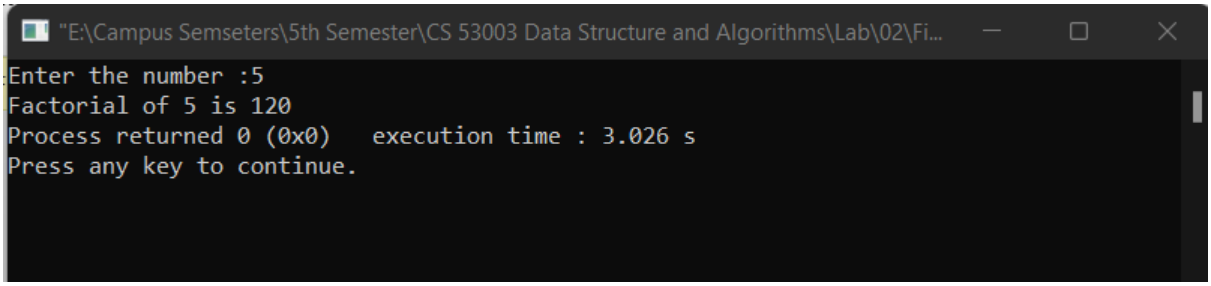
```cpp
#include<iostream>

using namespace std;

int Factorial(int m){
    if (m>1){
        return m*Factorial(m-1);
    }
}

int main(){
    int n;
    cout<<"Enter the number :";
    cin>>n;
    cout<<"Factorial of "<<n<<" is "<<Factorial(n);

}
```



3. Write a recursive function that computes the Fibonacci number corresponding to its positive integer argument, so that, for example, fibonacci(7) == 13.

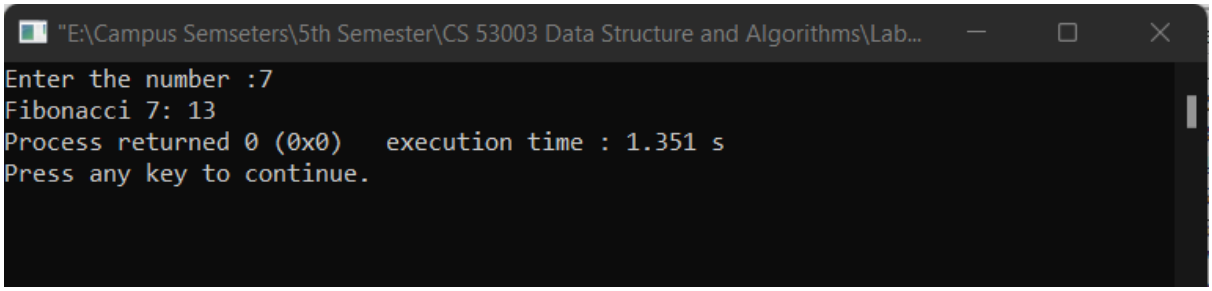
Start here | EX01.cpp | EX02part01.cpp | EX02part2.cpp | EX03.cpp

```cpp
#include<iostream>

using namespace std;

int Fibonacci(int m){
    int fib = 1;
    if (m>2){
        fib = Fibonacci(m-1) + Fibonacci(m-2);
        return fib;
    }
    else{
        return 1;
    }

}
int main(){
    int n;
    cout<<"Enter the number :";
    cin>>n;
    cout<<"Fibonacci "<<n<<": "<<Fibonacci(n);
}
```



4. Write C++ programs that use both recursive and non-recursive functions to find the GCD (greatest common divisor) of two given integers

Non-recursive

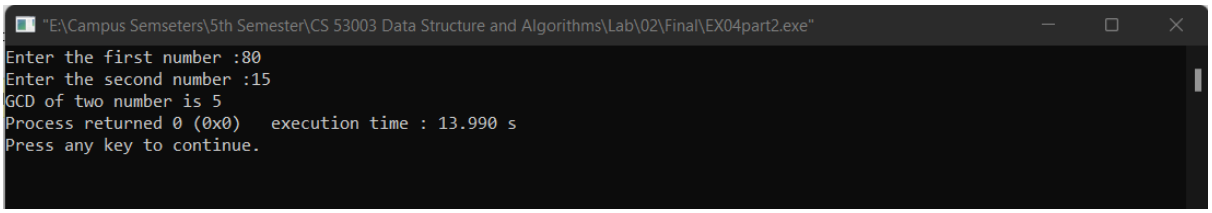
EX02part01.cpp | EX02part2.cpp | EX03.cpp | EX04part1.cpp | EX04part2.cpp

```cpp
#include<iostream>

using namespace std;

int main(){
    int a,b;
    cout<<"Enter the first number :";
    cin>>a;
    cout<<"Enter the second number :";
    cin>>b;

    while (a!=b){
       if (a>b){
            a=a-b;
       }
       else{
            b=b-a;
       }
    }
    cout<<"GCD of two number is "<<a;
}
```



Recursive

EX02part01.cpp | EX02part2.cpp | EX03.cpp | EX04part1.cpp | EX04part2.cpp

```cpp
#include<iostream>

using namespace std;

int GCD(int a,int b){
    if (a>b){
        GCD(a-b,b);
    }
    else if (b>a){
        GCD(a,b-a);
    }
    else if (a=b){
        return a;
    }

}
int main(){
    int n,m;
    cout<<"Enter the first number :";
    cin>>n;
    cout<<"Enter the second number :";
    cin>>m;
    cout<<"GCD of "<<n<<" & "<<m<<" is "<<GCD(n,m);
}
```

```
"E:\Campus Semseters\5th Semester\CS 53003 Data Structure and Algorithms\Lab...
Enter the first number :80
Enter the second number :15
GCD of 80 & 15 is 5
Process returned 0 (0x0)   execution time : 2.442 s
Press any key to continue.
```

5. Write a recursive function that finds and returns the minimum element in an array, where the array and its size are given as parameters.

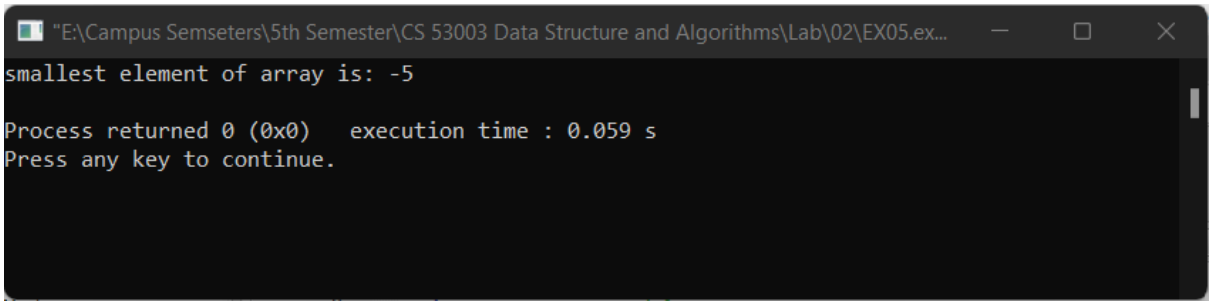
Start here | EX01.cpp | EX02part01.cpp | EX02part2.cpp | EX03.cpp | EX04part1.cpp | EX04part2.cpp | EX05.cpp

```cpp
#include <iostream>
using namespace std;

int findmin(int[],int size);

int main()
{
    int array[6]={74,10,23,785,-5,7};
    cout<<"smallest element of array is: " << findmin(array,5)<<endl;
    return 0;
}

int findmin(int array[],int counter)
{
    int smallest;
    if (counter==1)
        return array[0];
    else
    {
        smallest = findmin (array+1,counter-1);
        if (smallest<array[0])
            return smallest;
        else
        return array[0];
    }
}
```



***