**CSE115L – Programming Language I Lab**
**Lab – 23**
**Recursion**

In this lab, we will solve a few problems using recursive functions. Recursion is the process of defining a problem (or the solution to a problem) in terms of (a simpler version of) itself. For example, instead of defining the factorial of n as n! = 1*2*3*...*n, we can define it recursively as n! = n*(n-1)!. Every recursive definition must have a base case, where the definition is not recursive. In this example, the base case would be 0! = 1. Cases other than base cases are referred to as general cases.
The idea of recursion can be implemented in C using functions. Functions can solve problems recursively by calling themselves. Hence, these functions are referred to as recursive functions.

**Example 1: Write a C program that computes the factorial of a number using recursive function.**

```
#include<stdio.h>
int factorial(int x)
{
    if (n==1)
              return 1;
else
     return n*factorial(n-1);
}
```

```
void main()
{
    int N,i,result;
    printf("Enter N: \n");
    scanf("%d",&N);
    result = factorial(N);
    printf("%d! = %d",N,result);
}
```

**Example 2:** C program of a recursive function to find the sum of n natural numbers. For example: the number 5 will give an output of 15 since 5+4+3+2+1 = 15.

```
#include <stdio.h>
int sum(int n);
void main()
{
    int num,add;
    printf("Enter a positive integer:\n");
    scanf("%d",&num);
    add=sum(num);
    printf("sum=%d",add);
}
int sum(int n){
    if(n==0)
       return 0;
    else
       return n+sum(n-1);  /*self call  to function sum() */
}
```

**Example 3: Write a C program that computes the n-th Fibonacci number using recursive function.**

```
#include <stdio.h>

int fibonacci(int i) {

    if(i == 0) {
        return 0;
     }
```

```
     if(i == 1) {
          return 1;
      }

      return fibonacci(i-1) + fibonacci(i-2);

}


void main() {
      int n;
      scanf("%d", &n);
      printf("%d\n", fibonacci(n));
}
```

**Example 4: Write a C program to print all natural numbers from 1 to n using recursion.**

```
#include <stdio.h>
// Recursively prints all natural number between the given range.
void printNaturalNumber(int lowerLimit, int upperLimit)
{
    if(lowerLimit > upperLimit)
        return;

    printf("%d, ", lowerLimit);

    //Recursively calls the function to print next number
    printNaturalNumber(lowerLimit+1, upperLimit);
}

void main()
{
    int limit;

    printf("Print all natural numbers from 1 to : ");
    scanf("%d", &limit);

    printf("All natural numbers from 1 to %d are: ", limit);
    printNaturalNumber(1, limit);
}
```

**Perform the following tasks.**

**Task 1:** Write a C code to find the sum of the following series using RECURSIVE FUNCTIONS: $1^2 + 2^2 + 3^2 + \ldots + N^2$

**Task 2:** Write a C program using recursive function to find the inverse product of $1^{st}$ n natural numbers. E.g., for n= 5, the function should return (1/5) * (1/4) *(1/3) * (1/2) *(1/1) = 1/120 =0.00833