Lab's Scope:

- **Recursive Functions**

Problem 1:

Write a program that has a recursive function that calculates the sum of the first N natural numbers. For example if n=5 then the function should calculate 5+4+3+2+1.

Solution:

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

int nSum(int n)
{
    // base condition when N = 0
    if (n == 0) {
        return 0;
    }
    //recursive call
    int res = n + nSum(n - 1);
    return res;
}
////////////////////////////////////////
int main()
{
    int n = 5;
    int sum = nSum(n);

    cout << "Sum = " << sum;
    return 0;
}
```

Problem 2:

Write a program that has a recursive function ***findMin*** that given an array of positive integers, the function should calculate the minimum number in the array recursively.

Hint: you could use the C++ built-in function min to get the minimum between two numbers.

Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 7;

int findMin (int A[], int n)
{
    // if size = 0 means whole array has been traversed
    if (n == 1)
        return A[0];
    return min(A[n - 1], findMin (A, n - 1));
}
/////////////////////////////////////////////////////
int main()
{
    int A[] = { 3, 4, 65, 6, 50, 10, 2 };
    int n = SIZE;
    cout << "The minimum number is "<<findMin (A, n);
    return 0;
}
```

Problem 3:

Extend Problem number 2, to add the following functions:

- A recursive function ***findMax*** that calculates the maximum number in an array.
- A function ***findDiff*** that finds the difference between the maximum and minimum number of the array, by calling the two recursive functions of finding minimum and maximum.

Your main application should call the function ***findDiff.***

Hint: you could use the C++ built-in function max to get the maximum between two numbers.

Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 7;

int findMin(int A[], int n)
{
    // if size = 0 means whole array has been traversed
    if (n == 1)
        return A[0];
    return min(A[n - 1], findMin(A, n - 1));
}
/////////////////////////////////////////////////////
int findMax(int A[], int n)
{
    // if size = 0 means whole array has been traversed
    if (n == 1)
        return A[0];
    return max(A[n - 1], findMax(A, n - 1));
}
/////////////////////////////////////////////////////
int findDiff(int A[])
{
    int n = SIZE;
    int maximum = findMax(A, n);
    int minimum = findMin(A, n);
    return (maximum - minimum);
}
/////////////////////////////////////////////////////
int main()
{
    int A[] = { 3, 4, 65, 6, 50, 10, 2 };
    cout << "The Difference is "<<findDiff(A);
    return 0;
}
```

Problem 4:

Write a program that has a recursive function that prints the mathematical sequence that adds 5 to its previous term given that the first term is value 5. For example, if I need to print thill the tem 10, the outputs should be:

5 10 15 20 25 30 35 40 45 50

Solution:

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

int PrintSequence(int n)
{
    // base condition when n = 1
    if (n == 1) {
        cout << "5 ";
        return 5;
    }
    //recursive call
    int res = 5 + PrintSequence(n - 1);
    cout << res << " ";
    return res;
}
////////////////////////////////////////
int main()
{
   PrintSequence(10);
   return 0;
}
```