Assignment Day 6 | 30th December 2020

Question 1

Write a function to find the maximum element in the stack.

Sol:

```
using System;
using System.Collections.Generic;
class GfG
{
public class StackWithMax
{
static Stack<int> mainStack = new Stack<int> ();
static Stack<int> trackStack = new Stack<int> ();
public void push(int x)
    {
        mainStack.Push(x);
        if (mainStack.Count == 1)
        {
            trackStack.Push(x);
            return;
        }
if (x > trackStack.Peek())
```

```
            trackStack.Push(x);
        else
            trackStack.Push(trackStack.Peek());
    }

    public int getMax()
    {
        return trackStack.Peek();
    }
public void pop()
    {
        mainStack.Pop();
        trackStack.Pop();
    }
};
public static void Main()
{
    StackWithMax s = new StackWithMax();
    s.push(20);
    Console.WriteLine(s.getMax());
    s.push(10);
    Console.WriteLine(s.getMax());
    s.push(50);
    Console.WriteLine(s.getMax());
}
}
```

Question 2.

Write a function to find the minimum element in the stack.

Sol:

```
#include <iostream>
#include <stack>

class Stack
{
    std::stack<int> s;
 public:
    void push(int x)
    {
        if (s.empty()) {
            s.push(x);
            min = x;
        }
        else if (x > min) {
            s.push(x);
        }
        else {
            s.push(2 * x - min);
            min = x;
        }
    }
    void pop()
    {
        if (s.empty()) {
            std::cout << "Stack underflow!!" << '\n';
        }

        int top = s.top();
        if (top < min)
            min = 2 * min - top;
        s.pop();
    }
    int minimum()
    {
        return min;
    }
};
```

```
int main()
{
    Stack s;

    s.push(6);
    std::cout << s.minimum() << '\n';

    s.push(7);
    std::cout << s.minimum() << '\n';

    s.push(5);
    std::cout << s.minimum() << '\n';

    s.push(3);
    std::cout << s.minimum() << '\n';

    s.pop();
    std::cout << s.minimum() << '\n';

    s.pop();
    std::cout << s.minimum() << '\n';

    return 0;
}
```