Assignment 5 | 8th January 2021

Question 1

Write the function for insertion sort.

```
#include<iostream>
using namespace std;
int main ()
{
   int myarray[10] = { 12,4,3,1,15,45,33,21,10,2};

   cout<<"\nInput list is \n";
   for(int i=0;i<10;i++)
   {
      cout <<myarray[i]<<"\t";
   }
   for(int k=1; k<10; k++)
   {
      int temp = myarray[k];
      int j= k-1;
      while(j>=0 && temp <= myarray[j])
      {
         myarray[j+1] = myarray[j];
         j = j-1;
      }
      myarray[j+1] = temp;
   }
   cout<<"\nSorted list is \n";
   for(int i=0;i<10;i++)
   {
```

```
        cout <<myarray[i]<<"\t";
    }
}
```

Question 2

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

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
void findMax(stack<int> s)
{
    int m = s.top();
    int a;
    while (!s.empty())
    {
        a = s.top();
        if (m < a)
            m = a;
        s.pop();
    }
    cout << "\n\nThe maximum element of the Stack is: " << m << endl;
}
void show(stack<int> s)
{
    while (!s.empty())
    {
        cout << "  " << s.top();
        s.pop();
    }
    cout << endl;
}
int main()
{
    int i;
    stack<int> s;
    s.push(4);
    s.push(2);
```

```
    s.push(20);
    s.push(12);
    s.push(52);
    s.push(14);
    cout << "\n\nThe elements of the Stack in LIFO order are: ";
    show(s);
    findMax(s);
    cout << "\n\n\n";
    return 0;
}
```

Question 3

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

```
#include <bits/stdc++.h>
using namespace std;
struct MyStack
{
    stack<int> s;
    int minEle;
    void getMin()
    {
        if (s.empty())
            cout << "Stack is empty\n";
        else
            cout <<"Minimum Element in the stack is: "
                 << minEle << "\n";
    }
    void peek()
    {
        if (s.empty())
        {
            cout << "Stack is empty ";
            return;
        }
        int t = s.top();
```

```
        cout << "Top Most Element is: ";
        (t < minEle)? cout << minEle: cout << t;
    }
    void pop()
    {
        if (s.empty())
        {
            cout << "Stack is empty\n";
            return;
        }
        cout << "Top Most Element Removed: ";
        int t = s.top();
        s.pop();
        if (t < minEle)
        {
            cout << minEle << "\n";
            minEle = 2*minEle - t;
        }
        else
            cout << t << "\n";
    }
    void push(int x)
    {
        if (s.empty())
        {
            minEle = x;
            s.push(x);
            cout <<  "Number Inserted: " << x << "\n";
            return;
        }
        if (x < minEle)
        {
            s.push(2*x - minEle);
            minEle = x;
        }
        else
          s.push(x);
        cout <<  "Number Inserted: " << x << "\n";
    }
};
```

```
int main()
{
   MyStack s;
   s.push(3);
   s.push(5);
   s.getMin();
   s.push(2);
   s.push(1);
   s.getMin();
   s.pop();
   s.getMin();
   s.pop();
   s.peek();
   return 0;
}
```