**Assignment 5 | 8th January 2021**

**Question 1**

**Solution 1**

void insertionSort(int arr[], int n)

{

int i, key, j;

for (i = 1; i < n; i++) {

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

**Question 2**

**Solution 2**

void push(int x)

    {

        mainStack.push(x);

        if (mainStack.size() == 1)

        {

            trackStack.push(x);

            return;

        }

        if (x > trackStack.top())

         trackStack.push(x);

        else

            trackStack.push(trackStack.top());

    }

    int getMax()

    {

        return trackStack.top();

    }

    int pop()

    {

        mainStack.pop();

        trackStack.pop();

}

**Question 3**

**Solution 3**

using System;

using System.Collections;

public class MyStack

{

public Stack s;

public int minEle;

public MyStack()

{

s = new Stack();

}

public void getMin()

{

if (s.Count==0)

Console.WriteLine("Stack is empty");

else

Console.WriteLine("Minimum Element in the " +

" stack is: " + minEle);

}

public void Peek()

{

if (s.Count==0)

{

Console.WriteLine("Stack is empty ");

return;

}

int t =(int)s.Peek();

Console.Write("Top Most Element is: ");

if (t < minEle)

Console.WriteLine(minEle);

else

Console.WriteLine(t);

}

public void Pop()

{

if (s.Count==0)

{

Console.WriteLine("Stack is empty");

return;

}

Console.Write("Top Most Element Removed: ");

int t = (int)s.Pop();

if (t < minEle)

{

Console.WriteLine(minEle);

minEle = 2\*minEle - t;

}

else

Console.WriteLine(t);

}

public void Push(int x)

{

if (s.Count==0)

{

minEle = x;

s.Push(x);

Console.WriteLine("Number Inserted: " + x);

return;

}

if (x < minEle)

{

s.Push(2 \* x - minEle);

minEle = x;

}

else

s.Push(x);

Console.WriteLine("Number Inserted: " + x);

}

}