**Assignment Day 6 | 30th December 2020**

**Question-1:**

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

**Answer-1:**

struct MyStack {

    stack<int> s;

    int max\_ele;

     void getMax()

    {

        if (s.empty ())

            cout << "Stack is empty\n";

         else

            cout << "Maximum Element in the stack is: "

                 << max\_ele << "\n";

    }

    void peek()

    {

        if (s.empty()) {

            cout << "Stack is empty ";

            return;

        }

         int t = s.top();

        cout << "Top Most Element is: ";

         (t > max\_ele) ? cout << max\_ele : 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 > max\_ele)

{

            cout << max\_ele << "\n";

            max\_ele = 2 \* max\_ele - t;

        }

         else

            cout << t << "\n";

     }

    void push(int x)

    {

           if (s.empty())

{

            max\_ele = x;

            s.push(x);

            cout << "Number Inserted: " << x << "\n";

            return;

        }

          if (x > max\_ele)

{

            s.push(2 \* x – max\_ele);

            max\_ele = x;

        }

   else

            s.push(x);

        cout << "Number Inserted: " << x << "\n";

    }

}

**Question-2:**

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

**Answer-2:**

struct MyStack {

    stack<int> s;

    int min\_ele;

     void getmin()

    {

        if (s.empty ())

            cout << "Stack is empty\n";

         else

            cout << "Minimum Element in the stack is: "

                 << min\_ele << "\n";

    }

    void peek()

    {

        if (s.empty())

{

            cout << "Stack is empty ";

            return;

        }

         int t = s.top();

        cout << "Top Most Element is: ";

         (t > min\_ele) ? cout << min\_ele : 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 > min\_ele)

{

            cout << min\_ele << "\n";

            min\_ele = 2 \* min\_ele - t;

        }

         else

            cout << t << "\n";

     }

    void push(int x)

    {

           if (s.empty())

{

            min\_ele = x;

            s.push(x);

            cout << "Number Inserted: " << x << "\n";

            return;

        }

          if (x > min\_ele)

{

            s.push(2 \* x – min\_ele);

            max\_ele = x;

        }

   else

            s.push(x);

        cout << "Number Inserted: " << x << "\n";

    }

}