30/9/2020

Q Implement Stack:-

⇒ A stack can be implemented by means of Array, Structure, Pointer and Linked list. Stack can either be a fixed size one or it may have a sense of dynamic resizing.

Q. Program to find max. and minimum in an implement stack?

⇒

```
#include <stdio.h>
struct myStack.
{
    stack < int > s;
    int minEle;
    void getMin()
    {
        if ( s.empty() )
            count << "Stack is empty\n";
        else
            count << "Min element in
                the stack";
```

```
        << minEle << "\n";
}
void peek()
{
    if (s.empty())
    {
        count << "stack is empty";
        return;
    }
    int t = s.top();
    count << Top most element is: ";
}
void pop()
{
    if (s.empty())
    {
        count << "Stack is empty\n";
        return;
    }
    count << " Top most element remove";
    int t = s.top();
    s.pop();
    if (t < minEle)
    {
        count << minele << "\n";
        minEle = 2 * minele - t;
    }
```

```
        else
            count<<t << "\n";
    }
    void push (int x)
    {
        if (s.empty())
        {
            min ele = x;
            s.push(x);
            count << No. inserted: " << x<<
                                    "\n";
            return;
        }

        if (x < min ele)
        {
            s.push(2* x - min ele);
            min ele = x;
        }
        else
            s.push(x);
            cout << "No. inserted:" <<x<<
                                    "\n";
        }
    };
```

* Prog. for Maximum.

```
# include <stdio.h>
using namespace std;
struct MyStack {
    stack <int> s;
    int maxEle;
    void getMax()
    {
        if (s.empty())
            count << "Stack is empty\n";
        else
            count << "Max. Element in
                    stack is: "
                << maxEle << "\n";
    }
    void peek()
    {
        if (s.empty()) {
            count << "Stack is empty ";
            return;
        }
        int t = s.top();
        count << "top most element is: ";
    }
```

```
void pop()
{
    if (s.empty()) {
        count << "stack is empty\n";
        return;
    }
    count << "Top Most element remove:";
    int t = s.top();
    s.pop();

    if (t > Max ele) {
        count << max ele << "\n";
        max ele = 2 * max ele - t;
    }
    else
        count << t << "\n";
}
void push (int x)
{
    if (s.empty()) {
        max ele = x;
        s.push();
        count << "No. inserted: " << x << "\n";
        return;
    }
    if (x > max ele)
{
```

```
S.push(2*x - maxEle);
    maxEle = x;
}

else
    S.push(x);

cout << "No inserted:" << x << "\n";
}

};
```