Get minimum element from stack

You are given N elements and your task is to Implement a Stack in which you can get a minimum element.

By solving the above ques (Get min at pop) , you can solve this problem easily by creating a supporting stack , which can store the minimum elements. But we will try to solve this question in O(1) time complexity .

So , for O(1) time complexity , we cannot use any containers , and hence we will define a variable to store the minimum value.

ALGORITHM:

push(x):

1. Define a variable minEle to store minimum value.
2. If the stack is empty , then push x in the stack and minEle = x;
3. If the stack is not empty , then either x > stack.top() or x < stack.top().
4. If x > minEle , then push x in the stack .
5. If x < minEle , then minEle = x and push (2*x - stack.top()) in the stack , so that we can keep the track of minimum element before x.

pop():

1. As and when an element is popped out from the stack , two cases arise , either the popped element was the minimum element , or it was not .
2. If the popped element is less than minEle , then minEle = (2*minEle - tp) , where tp is the popped out element . Hence , we can retrieve the previous minimum element.

CODE :

```
1. #include <bits/stdc++.h>
2. using namespace std;

3.

4. struct MyStack
```

```
5.
{

6.      stack<int> s;

7.      int minEle;

8.      void getMin()
9.      {

10.          if (s.empty())

11.              cout << "Stack is empty\n";

12.          else

13.              cout <<"Minimum Element in the stack is: " <<
   minEle << endl;

14.     }

15.     void pop()

16.     {

17.          if (s.empty())

18.          {

19.              cout << "Stack is empty" << endl;

20.              return;

21.          }
```

```
22.
23.             cout << "Ele ment Popped : ";
24.             int tp = s.t op();
25.             s.pop();
26.             if (tp < min Ele)
27.             {
28.                 cout <<  'Element Popped : "<< minEle << endl;
29.                 minEle = 2*minEle - tp;
30.             }
31.             else
32.                 cout <<  p << endl;
33.     }
34.
35.     void push(int x)
36.     {
37.             if (s.empty())
38.             {
39.                 minEle = x;
40.                 s.push(x);
```

```
41.                 cout << "Element Pushed : " << x << endl;
42.                 return;
43.             }
44.             if (x < minEle)
45.             {
46.                 s.push(2*x - minEle);
47.                 minEle = x;
48.             }
49.             else
50.             s.push(x);
51.             cout << "Element Pushed : "  << x <<endl;
52.     }
53. };
54.
55. int main()
56. {
57.     MyStack s;
58.     s.push(2);
59.     s.push(4);
60.     s.getMin();
61.     s.push(3);
62.     s.push(1);
63.     s.getMin();
64.     s.pop();
65.     s.getMin();
66.     s.pop();
67.      s.getMin();
68.     return 0;
69. }
70.
```