```
#include <iostream>

#include <stack>
class Stack
{
std::stack<int> s;
    int min;
public:
    void push(int x)
      {
       if (s.empty()) {
        s.push(x);
        min = x;
        }
        else if (x > min) {
        s.push(x);
        }
        else {
            s.push(2 * x - min);
            min = x;
      }
  }
     void pop()
     {
       if (s.empty()) {
        std::cout << "Stack underflow!!" << '\n';
      }
     int top = s.top();
     if (top < min)
     min = 2 * min - top;
     s.pop();
     s.pop();
    }
    int minimum()
    {
       return min;
    }
};
int main()
{
Stack s;
s.push(6);
std::cout << s.minimum() << '\n';
s.push(7);
std::cout << s.minimum() << '\n';
s.push(5);
std::cout << s.minimum() << '\n';
s.push(3);
std::cout << s.minimum() << '\n';
s.pop();
std::cout << s.minimum() << '\n';
s.pop();
std::cout << s.minimum() << '\n';
return 0;
}
```

```
#include <iostream>
#include<stack>

using namespace std;



class StackWithMax

{
    stack<int> mainStack;
    stack<int> trackStack;

public:

    void push(int x)

    {

        mainStack.push(x);

        if (mainStack.size() == 1)

        {

            trackStack.push(x);

            return;

        }
          if (x > trackStack.top())
```

```
      if (x > trackStack.top())

       trackStack.push(x);

    else

       trackStack.push(trackStack.top());

 }



 int getMax()

 {

    return trackStack.top();

 }



 int pop()

 {

    mainStack.pop();

    trackStack.pop();

 }
```

```
        trackStack.pop();

    }

};
int main()

{

    StackWithMax s;

    s.push(20);

    cout << s.getMax() << endl;

    s.push(10);

    cout << s.getMax() << endl;

    s.push(50);

    cout << s.getMax() << endl;

    return 0;

}
```