```java
package stacksqueues;

// Java program to implement a stack that supports
// getMinimum() in O(1) time and O(1) extra space.
import java.util.*;

// A user defined stack that supports getMin() in
// addition to push() and pop()
class MyStack
{
    Stack<Integer> s;
    Integer minEle;

    // Constructor
    MyStack() { s = new Stack<Integer>(); }

    // Prints minimum element of MyStack
    void getMin()
    {
        // Get the minimum number in the entire stack
        if (s.isEmpty())
            System.out.println("Stack is empty");

            // variable minEle stores the minimum element
            // in the stack.
        else
            System.out.println("Minimum Element in the " +
                    " stack is: " + minEle);
    }

    // prints top element of MyStack
    void peek()
    {
        if (s.isEmpty())
        {
            System.out.println("Stack is empty ");
            return;
        }

        Integer t = s.peek(); // Top element.

        System.out.print("Top Most Element is: ");

        // If t < minEle means minEle stores
        // value of t.
        if (t < minEle)
            System.out.println(minEle);
        else
            System.out.println(t);
    }

    // Removes the top element from MyStack
    void pop()
    {
        if (s.isEmpty())
        {
            System.out.println("Stack is empty");
            return;
        }

        System.out.print("Top Most Element Removed: ");
        Integer t = s.pop();

        // Minimum will change as the minimum element
        // of the stack is being removed.
        if (t < minEle)
        {
            System.out.println(minEle);
            minEle = 2*minEle - t;
        }

        else
            System.out.println(t);
    }

    // Insert new number into MyStack
    void push(Integer x)
    {
        if (s.isEmpty())
        {
            minEle = x;
            s.push(x);
            System.out.println("Number Inserted: " + x);
            return;
        }

        // If new number is less than original minEle
        if (x < minEle)
        {
            s.push(2*x - minEle);
            minEle = x;
        }

        else
            s.push(x);

        System.out.println("Number Inserted: " + x);
    }
};

// Driver Code
 class Main
{
    public static void main(String[] args)
    {
        MyStack s = new MyStack();
        s.push(3);
        s.push(5);
        s.getMin();
        s.push(2);
        s.push(1);
        s.getMin();
        s.pop();
        s.getMin();
        s.pop();
        s.peek();
    }
}
```