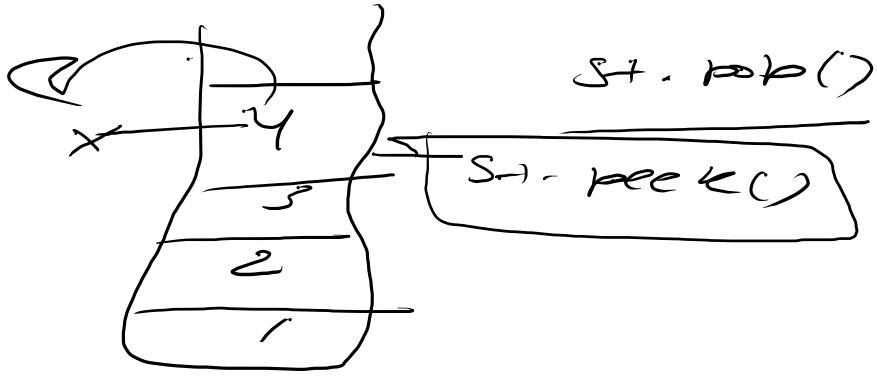
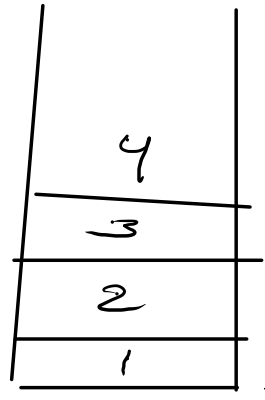
Stack<ClassName> objName = new Stack<>()



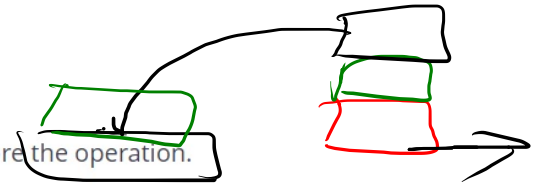
st.size()

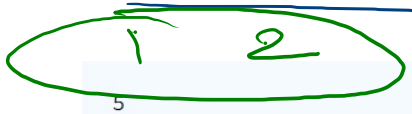
st.push()

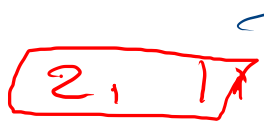


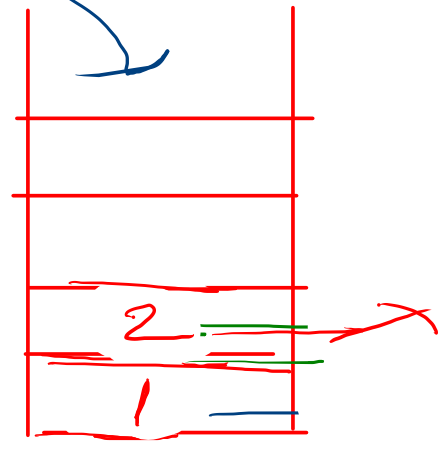
+push()
+pop()
+peek()
+size()
+search()

- push x, means push x in stack.
- pop means pop the element if stack is not empty, else ignore the operation.
- size means print the current size.
- display means print all the elements of stack.







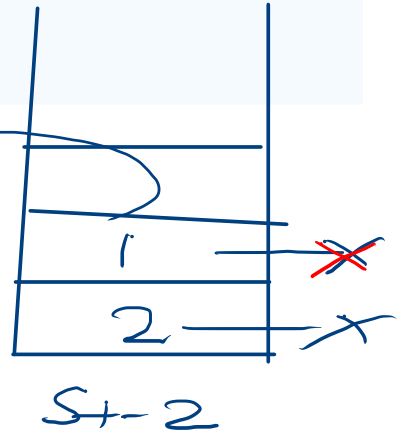


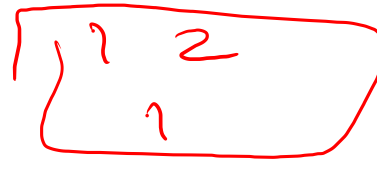

```
5
push 1
push 2
display
pop
size
```

int val = St.pop();

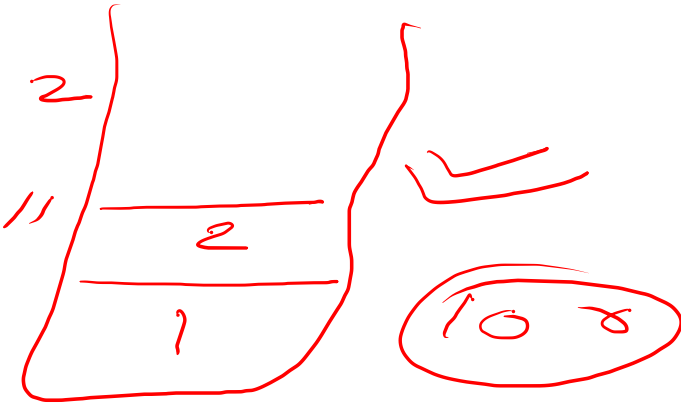
System.out.println(val);

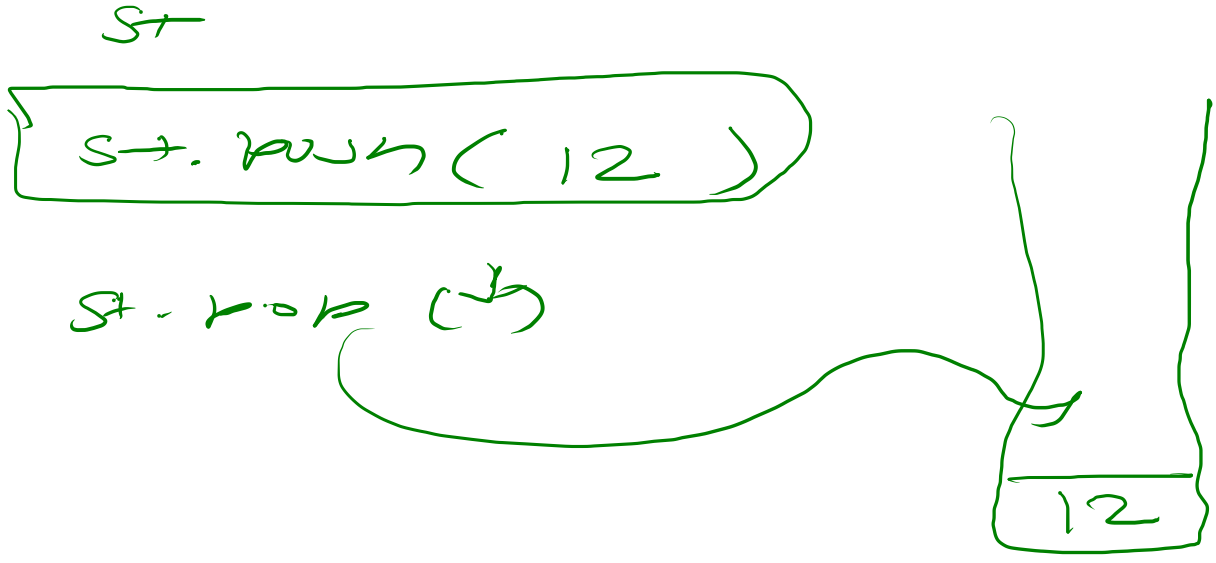
St.push(val);





```
int val1 = st.pop(); // 2
int val2 = st.pop(); // 1
SYSO(val2);
SYSO(val1);
st1.push(val2);
st1.push(val1);
```

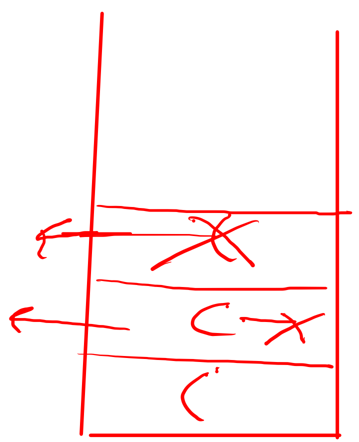


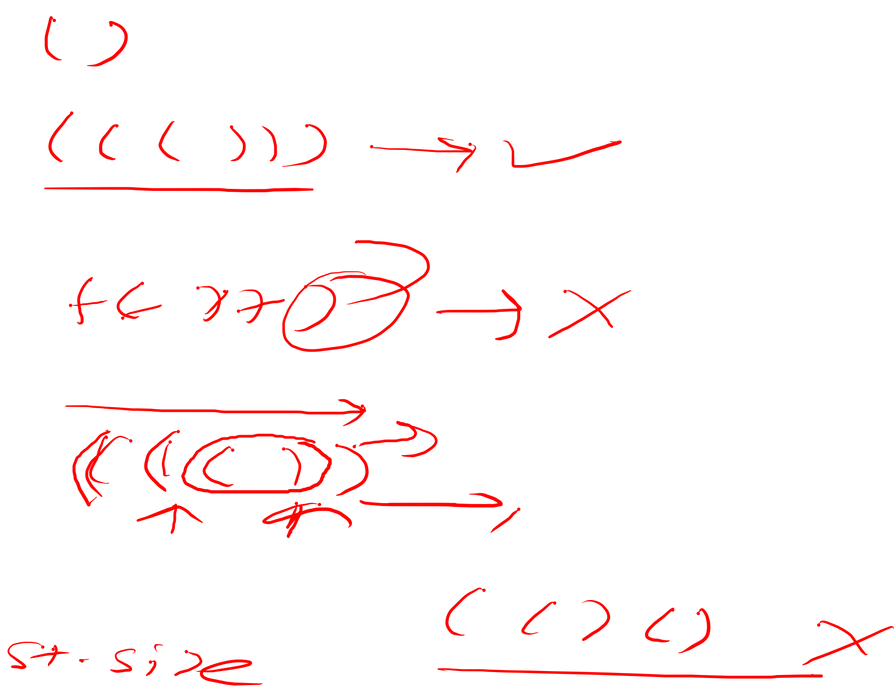

St
St.push(12)
St.pop()
12

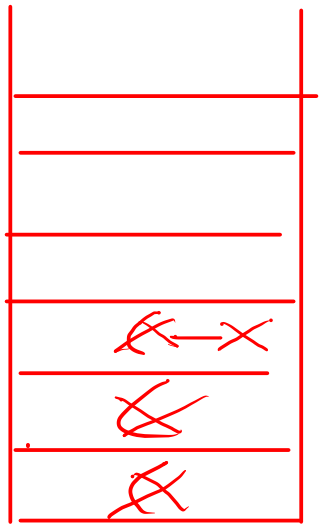
```
(()()
```

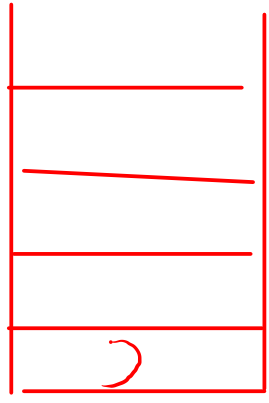
Sample Output 0

```
false
```

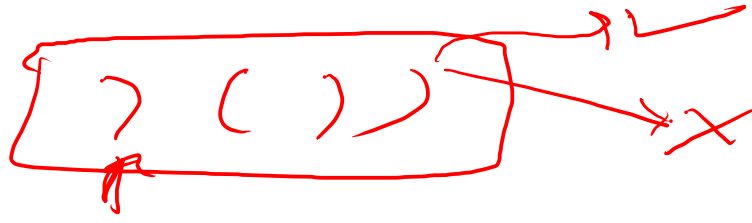








(( ) ( )) (( ) ((



if (( )

st.push();

else if(st.isEmpty())

return false;

Implement a stack using ArrayList

```java
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        // ArrayList<Integer> arr = new ArrayList<>();
        Stack<Integer> st = new Stack();
         Stack<Integer> st1 = new Stack();
        while(t-->0){
            String str = sc.next();
            if(str.equals("push")){
                int x = sc.nextInt();
                st.push(x);
            }else if(str.equals("pop")){
                if(st.size()>0){
                    st.pop();
                }
            }else if(str.equals("size")){
                System.out.println(st.size());
            }else if(str.equals("display")){
                while(st.size()>0){
                      st1.push(st.pop());
                }
                while(st1.size()>0){
                    int ans = st1.pop();
                    System.out.print(ans+" ");
                    st.push(ans);
                }
                System.out.println();
                // for(int i=0;i<arr.size();i++){
                //     st.push(arr.get(i));
                // }
                //  for(int i=arr.size()-1;i>=0;i--){
                //     System.out.print(arr.get(i)+" ");
                // }

                // System.out.println();
            }
        }

    }
}
```

Implement a stack using ArrayList

```java
public class Solution {
    static class StackUsingArrayList{
        List<Integer> stackList;
        // constructor
        StackUsingArrayList(){
            stackList = new Stack<>();
        }
        int length=0;
        // push
        public void push(int val){
            stackList.add(val);
            length++;
        }
        //pop
        int pop(){
            if(!isEmpty()){
                int popValue = stackList.get(stackList.size()-1);
                stackList.remove(stackList.size()-1);
                length--;
                return popValue;
            }else{
                System.out.println("The stack is empty");
                return -1;
            }
        }
        boolean isEmpty(){
            if(stackList.isEmpty()){
                return true;
            }else {
                return false;
            }
        }
        int display(){
            return stackList.get(stackList.size()-1);
        }
        int size(){
            return length;
        }
    }

```

```
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
     Scanner sc = new Scanner(System.in);
    int t = sc.nextInt();
   StackUsingArrayList st1 = new StackUsingArrayList();
    StackUsingArrayList st2 = new StackUsingArrayList();
    while(t-->0){
        String str = sc.next();
        if(str.equals("push")){
            int x = sc.nextInt();
            st1.push(x);
        }else if(str.equals("pop")){
            if(st1.size()>0){
                st1.pop();
            }
        }
        else if(str.equals("size")){
            System.out.println(st1.size());
        }
        else if(str.equals("display")){
            while(!st1.isEmpty()){
                st2.push(st1.pop());
            }
            while(!st2.isEmpty()){
                int val = st2.pop();
                System.out.print(val+" ");
                st1.push(val);
            }
            System.out.println();
        }
    }
        }
```

valid parentheses 10

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        boolean ans = isParenthesis(str);
        System.out.println(ans);
    }
    public static boolean isParenthesis(String str){
        Stack<Character> st = new Stack<>();
        for(int i=0;i<str.length();i++){
            char ch = str.charAt(i);
            if(ch=='('){
                st.push(ch);
            }else{
                if(st.isEmpty()){
                    return false;
                }
                else{
                    st.pop();
                }
            }
        }
        return st.isEmpty();
    }
}
```