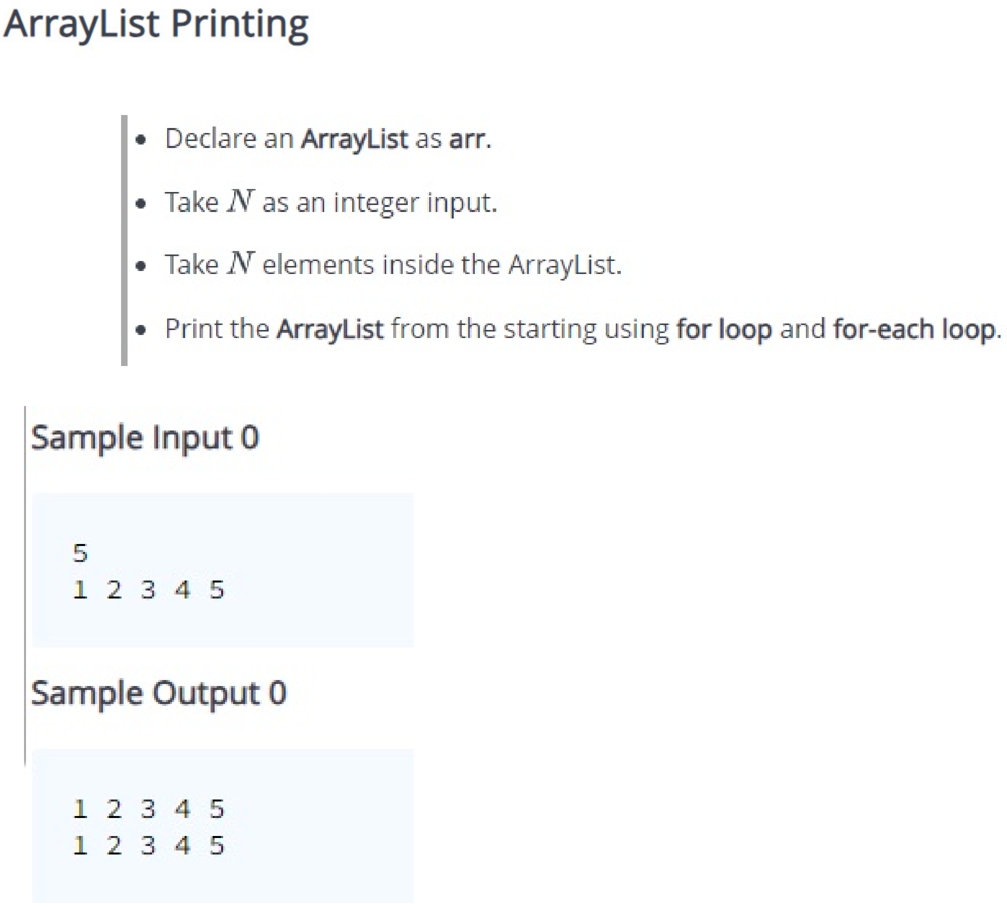


ArrayList Printing

> - Declare an **ArrayList** as **arr**.
> - Take N as an integer input.
> - Take N elements inside the ArrayList.
> - Print the **ArrayList** from the starting using **for loop** and **for-each loop**.

Sample Input 0

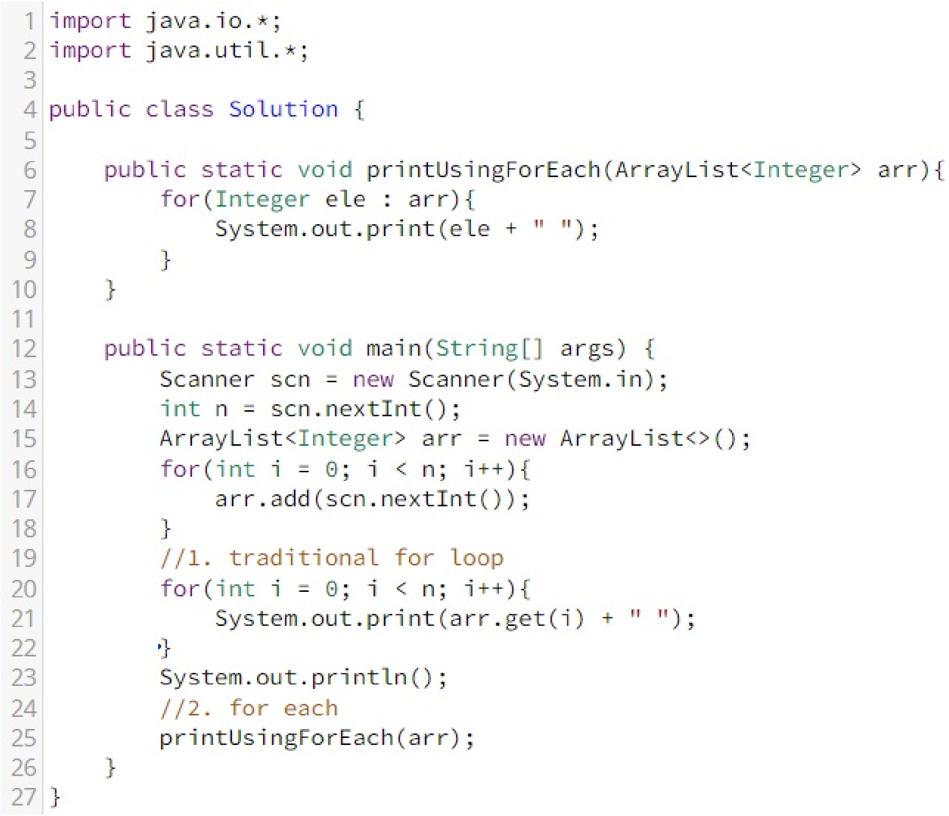
```
5
1 2 3 4 5
```

Sample Output 0

```
1 2 3 4 5
1 2 3 4 5
```

n

1 2 3 4 5

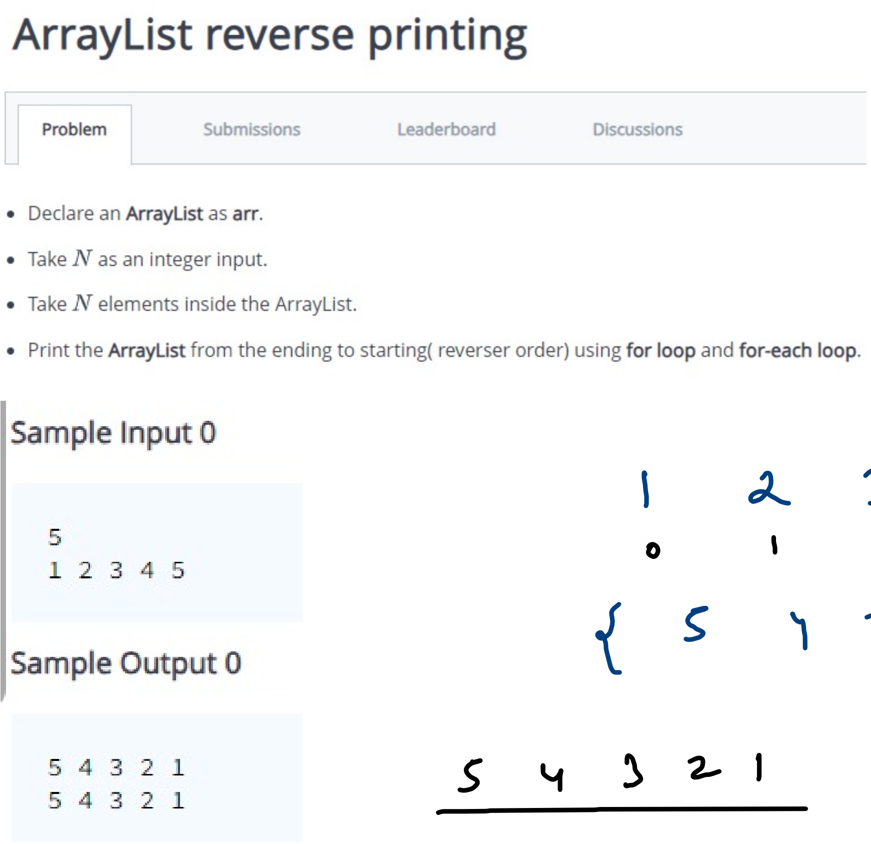


```
import java.io.*;
import java.util.*;

public class Solution {

    public static void printUsingForEach(ArrayList<Integer> arr){
        for(Integer ele : arr){
            System.out.print(ele + " ");
        }
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        ArrayList<Integer> arr = new ArrayList<>();
        for(int i = 0; i < n; i++){
            arr.add(scn.nextInt());
        }
        //1. traditional for loop
        for(int i = 0; i < n; i++){
            System.out.print(arr.get(i) + " ");
        }
        System.out.println();
        //2. for each
        printUsingForEach(arr);
    }
}
```



ArrayList reverse printing

- Declare an **ArrayList** as **arr**.
- Take N as an integer input.
- Take N elements inside the ArrayList.
- Print the **ArrayList** from the ending to starting(reverser order) using **for loop** and **for-each loop**.

Sample Input 0

```
5
1 2 3 4 5
```

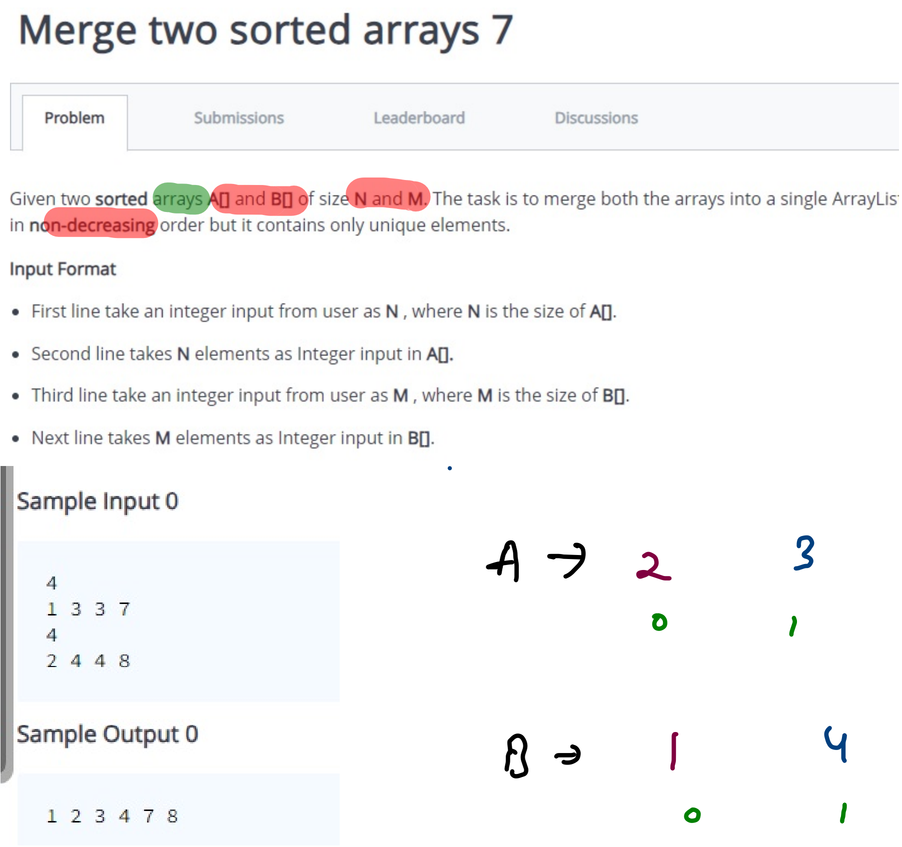
Sample Output 0

```
5 4 3 2 1
5 4 3 2 1
```

1 2 3 4 5
0 1 2 3 4

{ 5 4 3 2 1 }

5 4 3 2 1



Merge two sorted arrays 7

Given two **sorted** arrays A[] and B[] of size N and M. The task is to merge both the arrays into a single ArrayLis
in **non-decreasing** order but it contains only unique elements.

Input Format

- First line take an integer input from user as **N** , where **N** is the size of **A[]**.
- Second line takes **N** elements as Integer input in **A[]**.
- Third line take an integer input from user as **M** , where **M** is the size of **B[]**.
- Next line takes **M** elements as Integer input in **B[]**.

Sample Input 0

```
4
1 3 3 7
4
2 4 4 8
```

Sample Output 0

```
1 2 3 4 7 8
```

i

A → 2 3 3 7

0 1 2 3

j<n

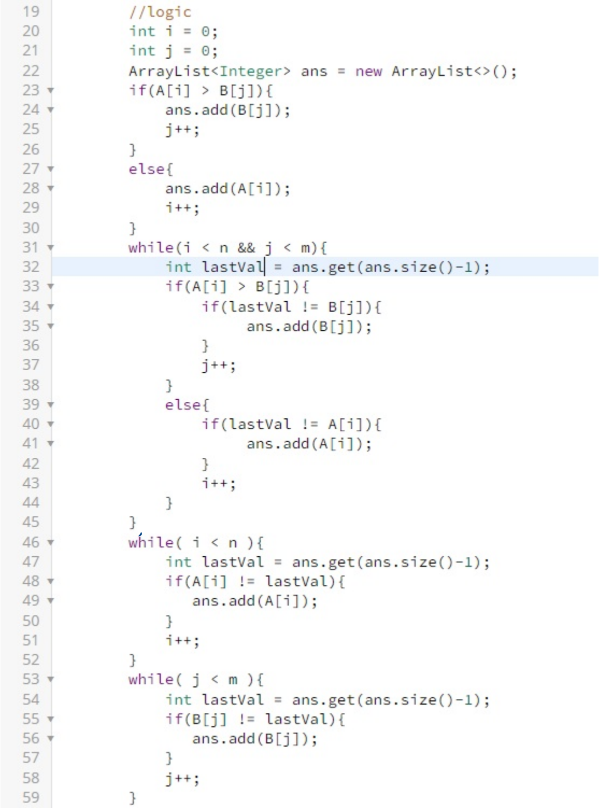
B → 1 4 4 8 9 9 10

0 1 2 3 4 5

j

lastVal != curr

{ 1 2 3 4 7 8 9 10 }



```
        //logic
        int i = 0;
        int j = 0;
        ArrayList<Integer> ans = new ArrayList<>();
        if(A[i] > B[j]){
            ans.add(B[j]);
            j++;
        }
        else{
            ans.add(A[i]);
            i++;
        }
        while(i < n && j < m){
            int lastVal = ans.get(ans.size()-1);
            if(A[i] > B[j]){
                if(lastVal != B[j]){
                    ans.add(B[j]);
                }
                j++;
            }
            else{
                if(lastVal != A[i]){
                    ans.add(A[i]);
                }
                i++;
            }
        }
        while( i < n ){
            int lastVal = ans.get(ans.size()-1);
            if(A[i] != lastVal){
               ans.add(A[i]);
            }
            i++;
        }
        while( j < m ){
            int lastVal = ans.get(ans.size()-1);
            if(B[j] != lastVal){
               ans.add(B[j]);
            }
            j++;
        }
```

A → 2 2 5 6 6

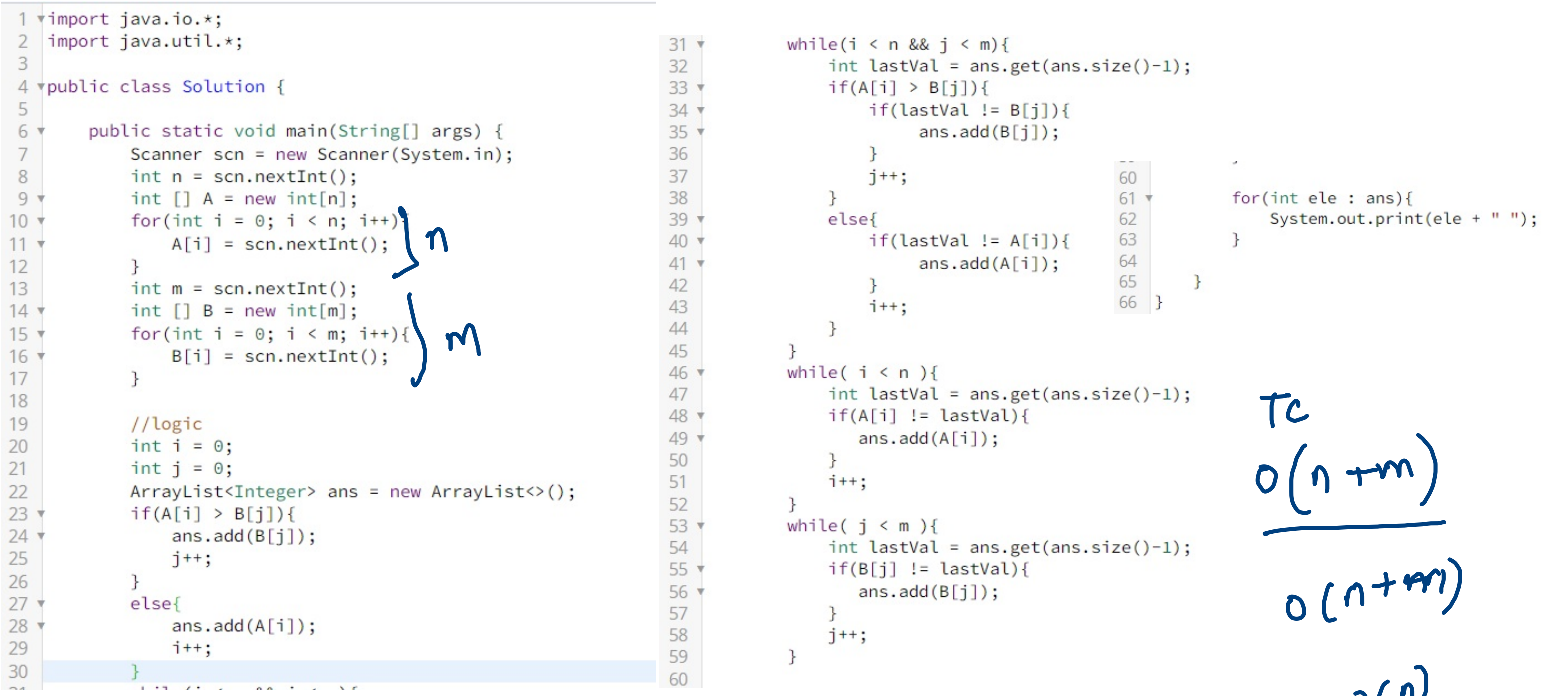
i

B → 1 3 4 4

ev = ~~2~~ ~~3~~ 4

j

{ 1 2 3 4 5 6 }



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        int m = scn.nextInt();
        int [] B = new int[m];
        for(int i = 0; i < m; i++){
            B[i] = scn.nextInt();
        }

        //logic
        int i = 0;
        int j = 0;
        ArrayList<Integer> ans = new ArrayList<>();
        if(A[i] > B[j]){
            ans.add(B[j]);
            j++;
        }
        else{
            ans.add(A[i]);
            i++;
        }
        while(i < n && j < m){
            int lastVal = ans.get(ans.size()-1);
            if(A[i] > B[j]){
                if(lastVal != B[j]){
                     ans.add(B[j]);
                }
                j++;
            }
            else{
                if(lastVal != A[i]){
                     ans.add(A[i]);
                }
                i++;
            }
        }
        while( i < n ){
            int lastVal = ans.get(ans.size()-1);
            if(A[i] != lastVal){
               ans.add(A[i]);
            }
            i++;
        }
        while( j < m ){
            int lastVal = ans.get(ans.size()-1);
            if(B[j] != lastVal){
               ans.add(B[j]);
            }
            j++;
        }

        for(int ele : ans){
            System.out.print(ele + " ");
        }
    }
}
```

n

m

TC

$\underline{O(n+m)}$

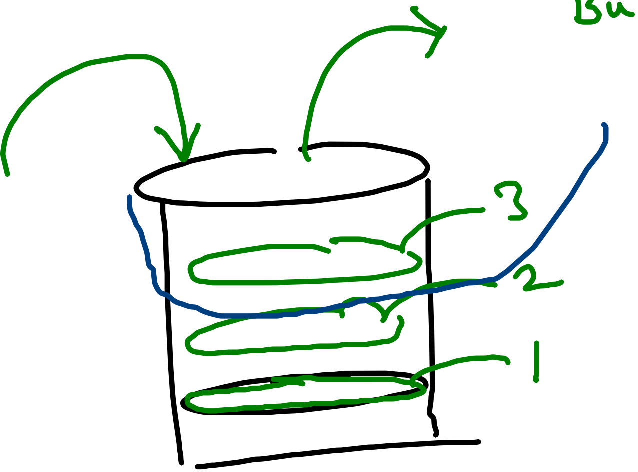
$O(n+m)$

$O(n)$.

D.S. → Stack

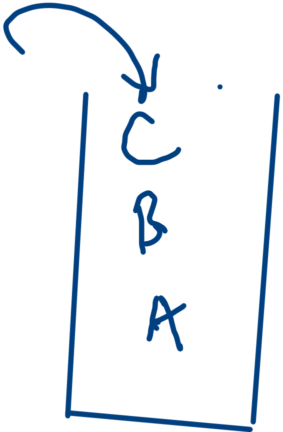
Bucket like.

Disk 1



LIFO

Last in first out



C

B

A

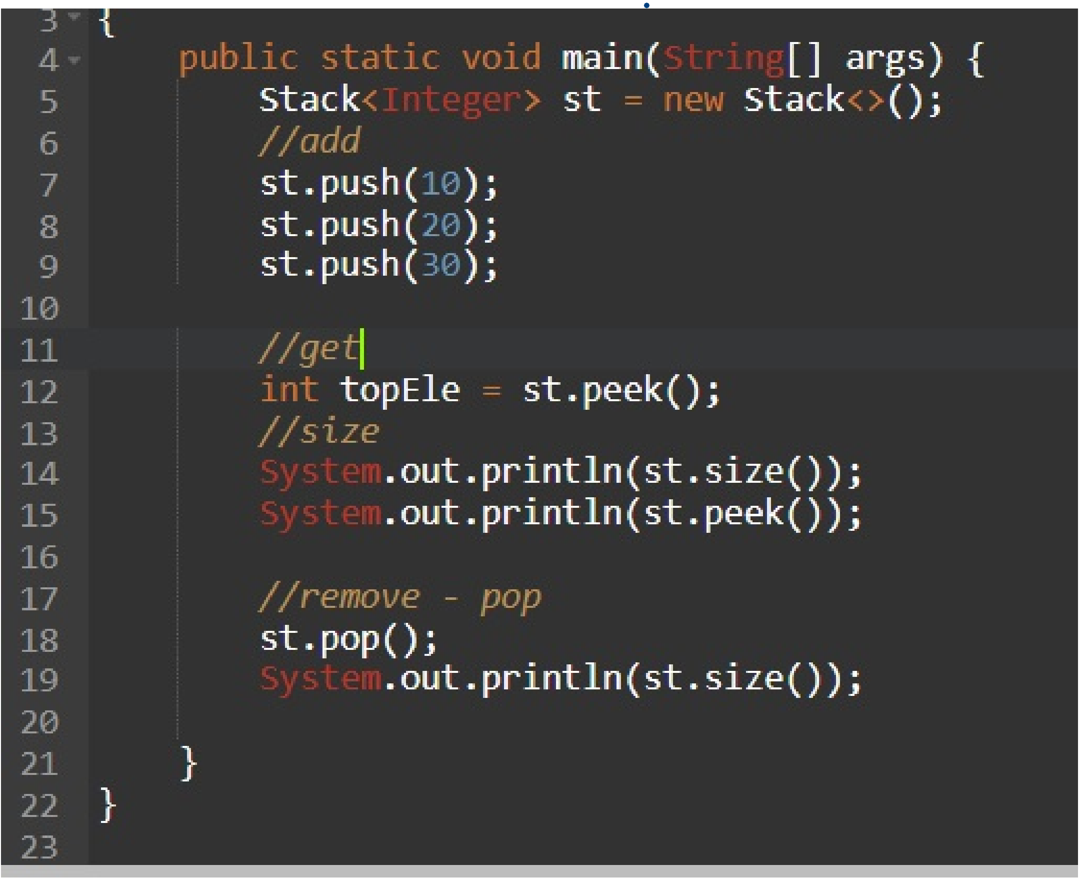
Initialize

add → push

get → peek

remove → pop

size → size



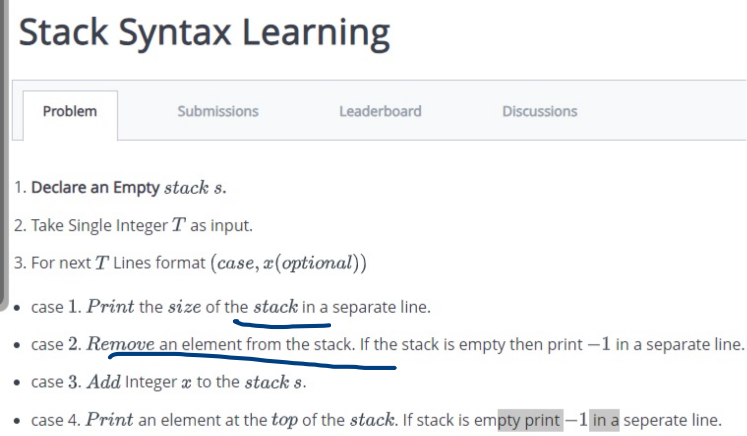
```
{
    public static void main(String[] args) {
        Stack<Integer> st = new Stack<>();
        //add
        st.push(10);
        st.push(20);
        st.push(30);

        //get
        int topEle = st.peek();
        //size
        System.out.println(st.size());
        System.out.println(st.peek());

        //remove - pop
        st.pop();
        System.out.println(st.size());

    }
}

```



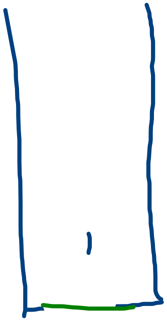
Stack Syntax Learning

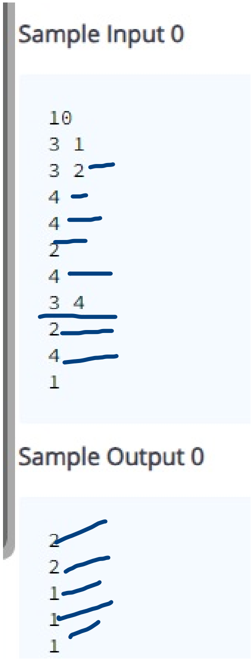
1. **Declare an Empty** $stack$ s.
2. Take Single Integer T as input.
3. For next T Lines format $(case, x(optional))$

- case 1. $Print$ the $size$ of the $stack$ in a separate line.
- case 2. $Remove$ an element from the stack. If the stack is empty then print -1 in a separate line.
- case 3. Add Integer x to the $stack$ s.
- case 4. $Print$ an element at the top of the $stack$. If stack is empty print -1 in a seperate line.

x

3 , 1



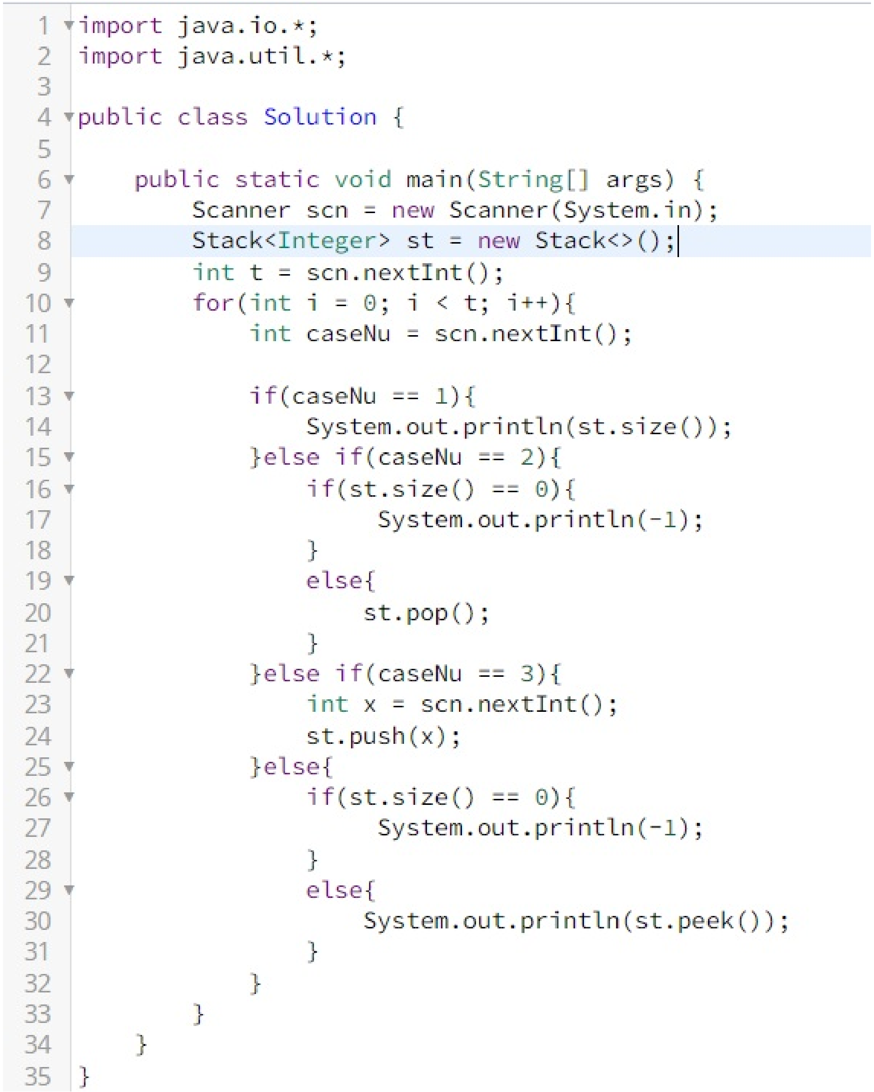


Sample Input 0

```
10
3 1
3 2
4
4
2
4
3 4
2
4
1
```

Sample Output 0

```
2
2
1
1
1
```



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        Stack<Integer> st = new Stack<>();
        int t = scn.nextInt();
        for(int i = 0; i < t; i++){
            int caseNu = scn.nextInt();

            if(caseNu == 1){
                System.out.println(st.size());
            }else if(caseNu == 2){
                if(st.size() == 0){
                     System.out.println(-1);
                }
                else{
                    st.pop();
                }
            }else if(caseNu == 3){
                int x = scn.nextInt();
                st.push(x);
            }else{
                if(st.size() == 0){
                     System.out.println(-1);
                }
                else{
                    System.out.println(st.peek());
                }
            }
        }
    }
}
```