

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 ArrayList in non-decreasing order but it contains only unique elements.

Sample Input 0

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

$N=5$

1	3	3	4	7
0	1	2	3	4

i

$\{1, 2, 3, 4, 7, 8, 9, 12\}$

Sample Output 0

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

$M=4$

2	4	4	8	9	12
0	1	2	3	4	5

j

smaller = 3,

smaller != last Ele
↳ add.

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static void main(String[] args) {
6         Scanner scn = new Scanner(System.in);
7         int n = scn.nextInt();
8         int [] A = new int[n];
9         for(int i = 0; i < n; i++){
10             A[i] = scn.nextInt();
11         }
12         int m = scn.nextInt();
13         int [] B = new int[m];
14         for(int i = 0; i < m; i++){
15             B[i] = scn.nextInt();
16         }
17         ArrayList<Integer> arr = new ArrayList<>();
18         int i = 0;
19         int j = 0;
20         if(A[i] > B[j]){
21             arr.add(B[j]);
22             j++;
23         }else{
24             arr.add(A[i]);
25             i++;
26         }
27         while(i < n && j < m){
28             //compare
29             int lastVal = arr.get(arr.size()-1);
30             int min = Math.min(A[i], B[j]);
31             if(min != lastVal){
32                 arr.add(min);
33             }
34             if(min == A[i]){
35                 i++;
36             }else{
37                 j++;
38             }
39         }

```

A → 1 2 2 5 7 8

i

B → 2 3 6 6

j

< 1, 2, 3, 5, 6, 7, 8 >

lv = 6

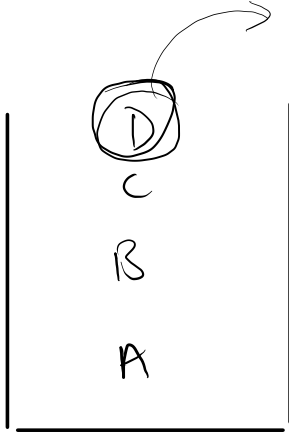
min = 6

```

while(j < m){
    int lastVal = arr.get(arr.size()-1);
    if(B[j] != lastVal){
        arr.add(B[j]);
    }
    j++;
}
while(i < n){
    int lastVal = arr.get(arr.size()-1);
    if(A[i] != lastVal){
        arr.add(A[i]);
    }
    i++;
}
for(int ele : arr){
    System.out.print(ele + " ");
}
}

```

Stack }
Bucket like → D.S.



LIFO → Last in first out

A B C D

Stack

- initialize
- size
- add push
- remove pop
- get peek

```
4
5 public class Main
6 {
7     public static void main(String[] args) {
8         //init
9         Stack<Integer> st = new Stack<>();
10        //add
11        st.push(10);
12        st.push(20);
13        st.push(30);
14
15        //size
16        System.out.println(st.size());
17
18        //remove: pop
19        System.out.println(st.peek());
20        st.pop();
21        System.out.println(st.peek());
22        System.out.println(st.size());
23    }
24 }
```

3
30
20
2

~~30~~
20
10

Stack Syntax Learning

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         Stack<Integer> st = new Stack<>();
9         int t = scn.nextInt();
10        while(t-- > 0){
11            int caseNu = scn.nextInt();
12            if(caseNu == 1){
13                System.out.println(st.size());
14            }else if(caseNu == 2){
15                if(st.size() == 0){
16                    System.out.println(-1);
17                }else{
18                    st.pop();
19                }
20            }else if(caseNu == 3){
21                st.push(scn.nextInt());
22            }else{
23                if(st.size() == 0){
24                    System.out.println(-1);
25                }else{
26                    System.out.println(st.peek());
27                }
28            }
29        }
30    }
31 }
```

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.

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

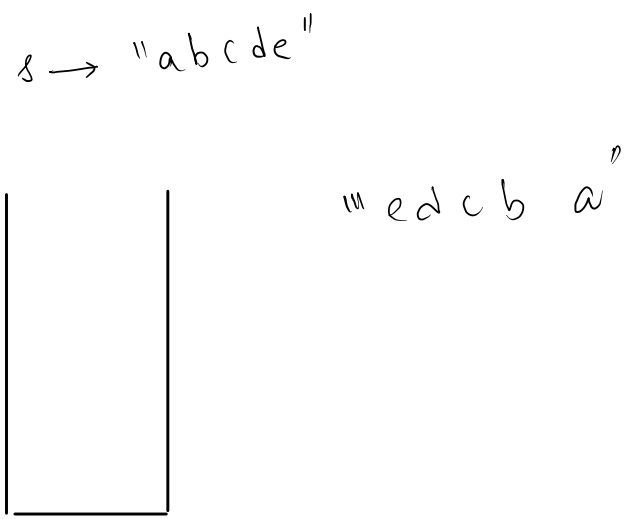
Sample Output 0

```
2
2
1
1
1
```

Reverse string

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Given a String *Str*. We have to *Reverse* the string *Str* with help of only *stacks*.



Language: Java 8

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         String s = scn.next();
9         Stack<Character> st = new Stack<>();
10
11         for(int i = 0; i < s.length(); i++){
12             st.push(s.charAt(i));
13         }
14         String ans = "";
15         //logic
16         while(st.size() != 0){
17             ans += st.pop();
18         }
19         System.out.println(ans);
20     }
21 }
```

Reverse Words in a Given String

$s \rightarrow$ " I am with Geekster "

Geekster
with
am
I

Geekster _ with _ am _ I

