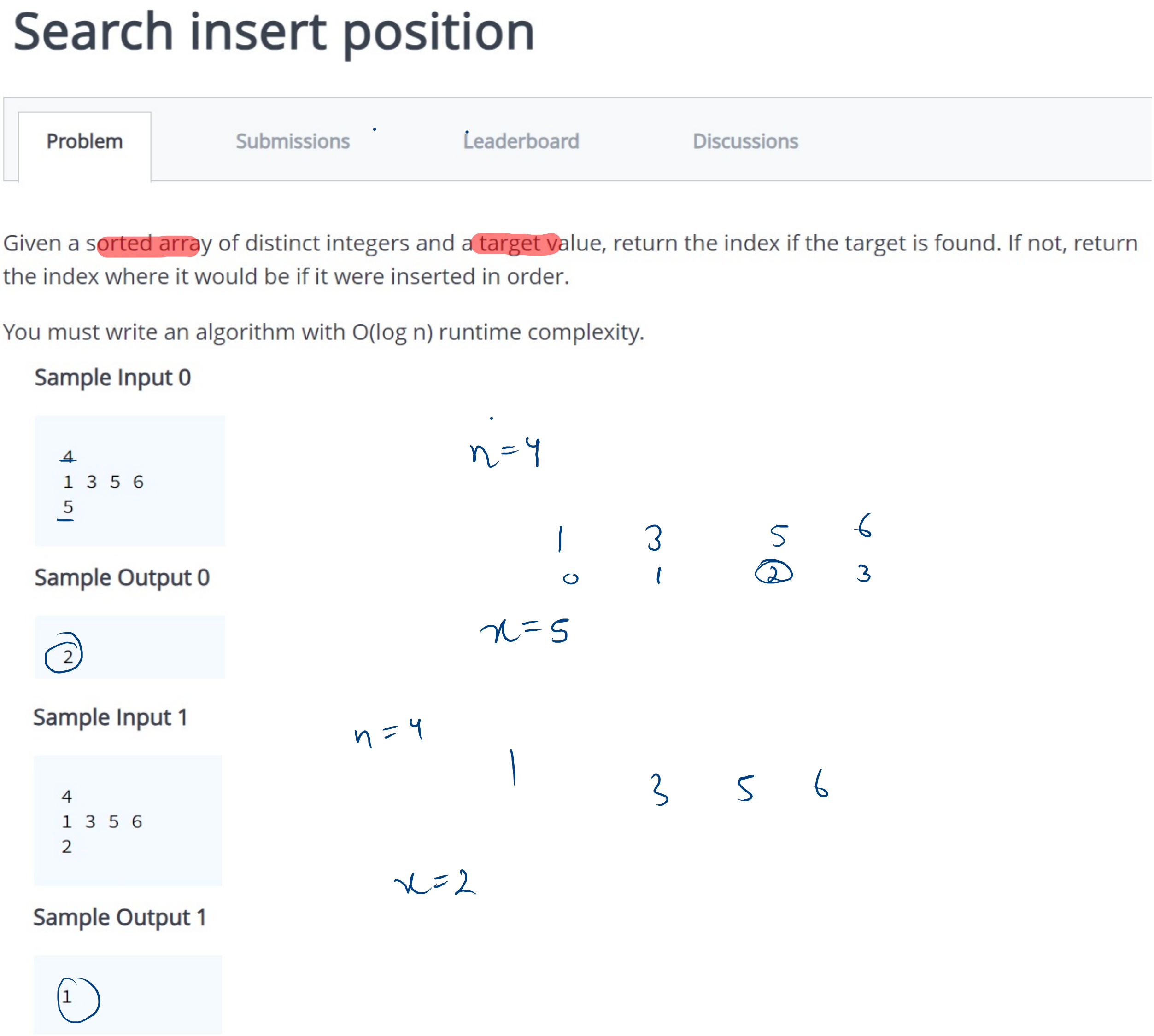


Search insert position

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You must write an algorithm with O(log n) runtime complexity.

Sample Input 0

```
4
1 3 5 6
5
```

n=4

1	3	5	6
0	1	2	3

x=5

Sample Output 0

```
2
```

Sample Input 1

```
4
1 3 5 6
2
```

n=4

1 3 5 6

x=2

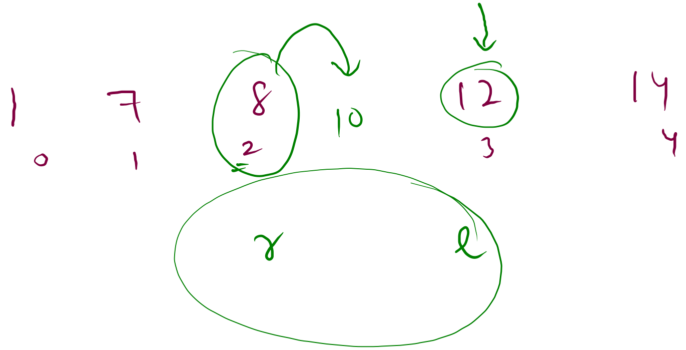
Sample Output 1

```
1
```

$\boxed{x = \quad 10}$

ans = $\underline{\underline{2}}$

?
ans = 3.



$l \leq r$

↳ $m = 2$

$8 \neq 10$

$8 \not< 10$ } $m' = 3+4/2 = 3.$

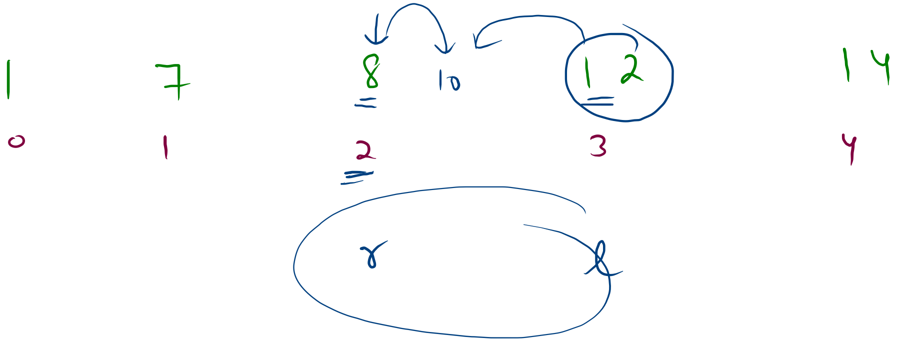
$10 \not\leq 12$

$10 > 12$

$10 < 12$

$x = 10$

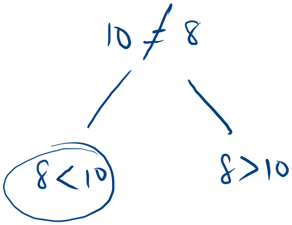
ans = ~~-1~~ 2



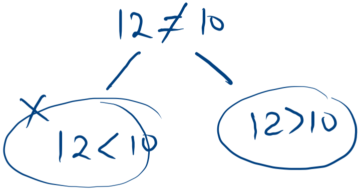
✓

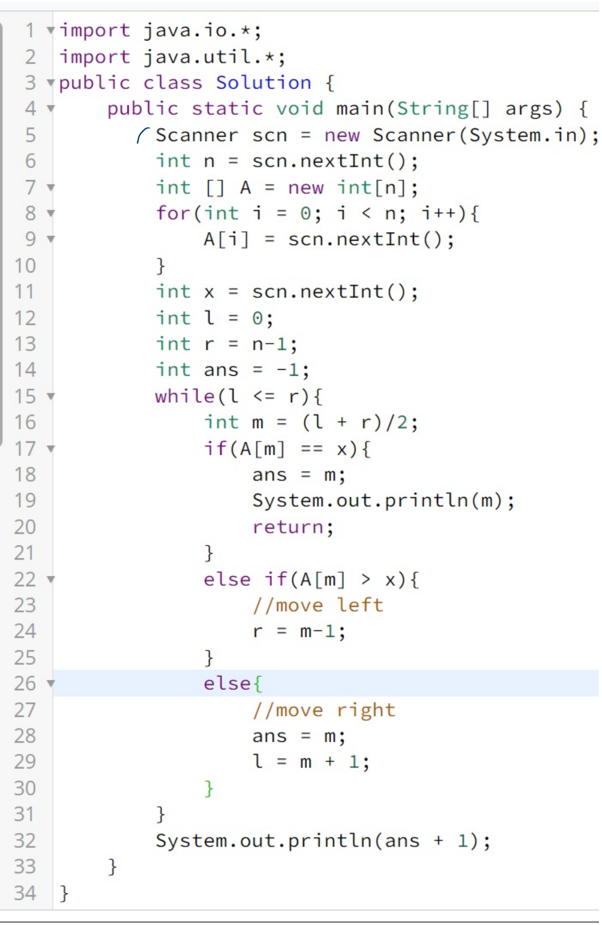
$l \le r$

$\hookrightarrow m_1 = 2$



$m_2 = 3$.





```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 x = scn.nextInt();
        int l = 0;
        int r = n-1;
        int ans = -1;
        while(l <= r){
            int m = (l + r)/2;
            if(A[m] == x){
                ans = m;
                System.out.println(m);
                return;
            }
            else if(A[m] > x){
                //move left
                r = m-1;
            }
            else{
                //move right
                ans = m;
                l = m + 1;
            }
        }
        System.out.println(ans + 1);
    }
}
```

$x = 2$

ans = ~~-1~~ 0

↓

1	3	5	6	7
0	1	2	3	4

r

l

$0 \le 4 \rightsquigarrow m = 2$

$5 == 2$ ✗

$5 > 2$ ✗

$1 \le 1$ ✓

$m = 1$

$3 == 2$

$3 > 2$

$0 \le 1 \rightsquigarrow m = (0+1)/2 = 0$

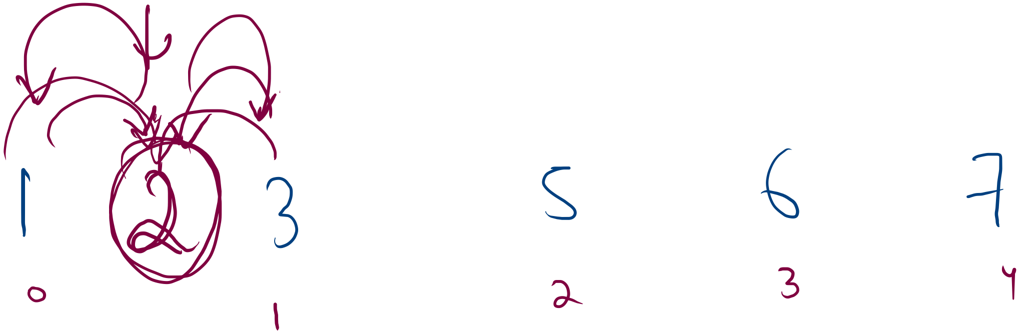
$1 == 2$ ✗

$1 > 2$ ✗

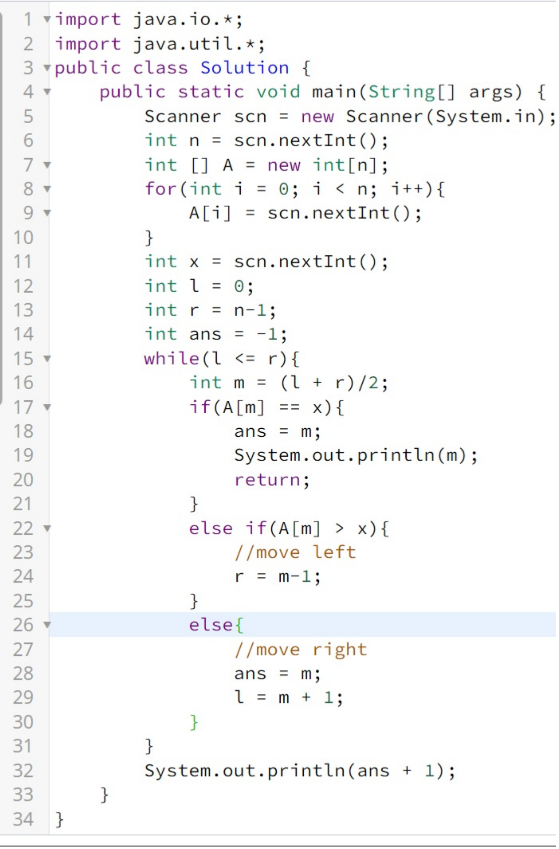
$x = 2.$

1	3	5	6	7
0	1	2	3	4

$x = 2$



idx = 1



```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 x = scn.nextInt();
        int l = 0;
        int r = n-1;
        int ans = -1;
        while(l <= r){
            int m = (l + r)/2;
            if(A[m] == x){
                ans = m;
                System.out.println(m);
                return;
            }
            else if(A[m] > x){
                //move left
                r = m-1;
            }
            else{
                //move right
                ans = m;
                l = m + 1;
            }
        }
        System.out.println(ans + 1);
    }
}
```

Key = 1

2 3 5 6

0 1 2 3

l

r

ans = -1

-1 +1 = 0

Key < A(0) → 0

0 ≤ 3

↳ m = 0+3/2 = 1

$3 == 1$ ✗

$3 > 1$

0 ≤ 0

m = 0

$2 == 1$ ✗

$2 > 1$ ✓

Arraylist → Dynamic D.S.

Collection.

Integer

↓

{ 1 2 3 4 5 6 7 8 }

To learn. D.S

Initialize. ✓

add ✓

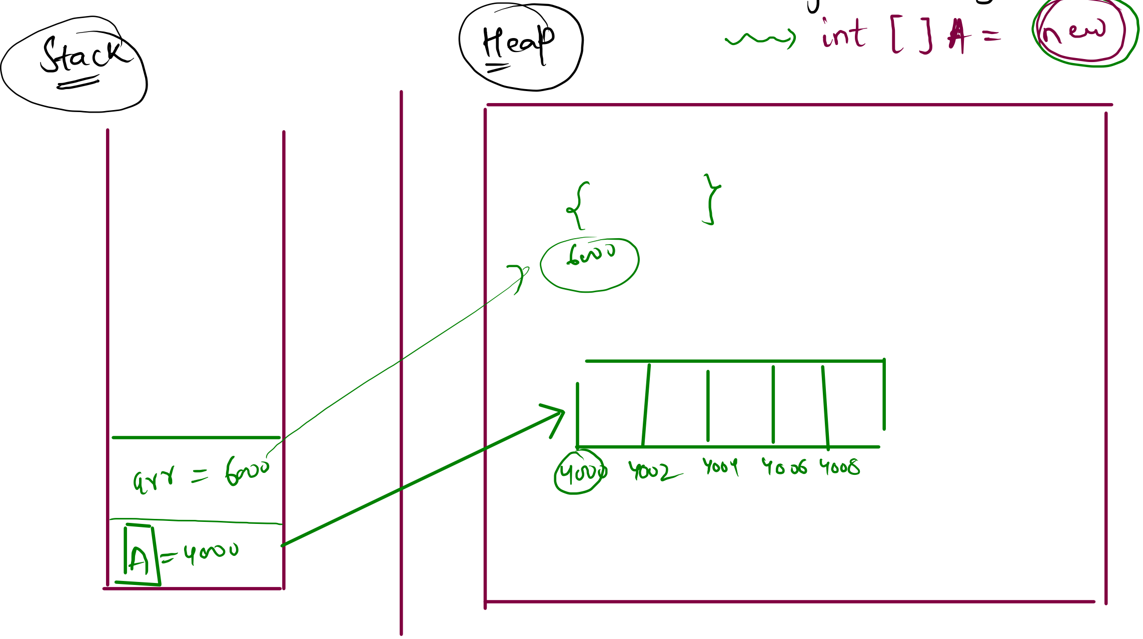
get ✓

size ✓

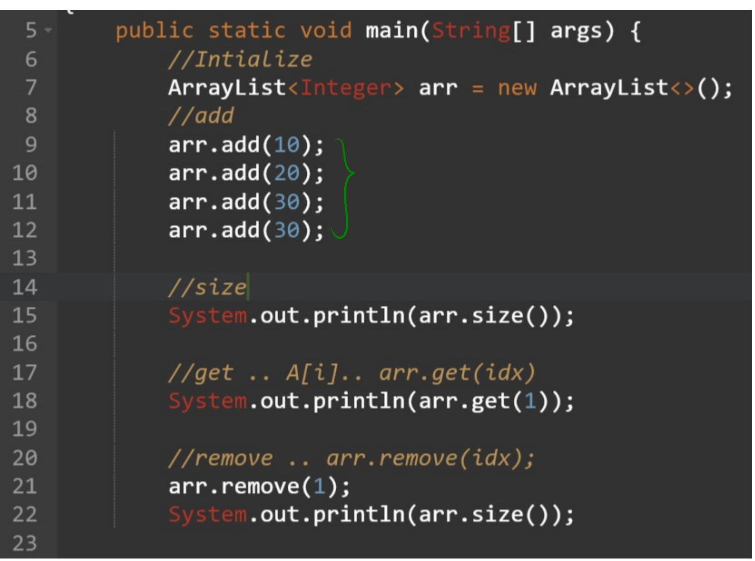
delete ✓

traverse / print. ✓

```
ArrayList < Integer> arr = new ArrayList<>();
~~> int [ ] A = new int[5];
```



* JVM



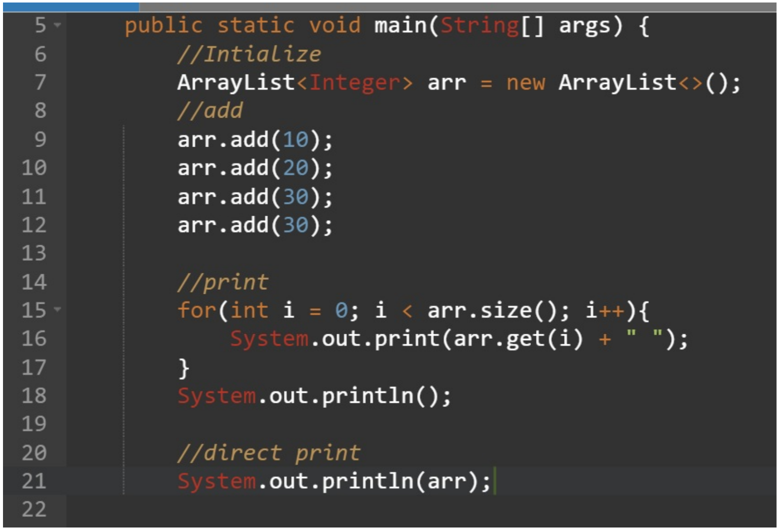
```java
    public static void main(String[] args) {
        //Intialize
        ArrayList<Integer> arr = new ArrayList<>();
        //add
        arr.add(10);
        arr.add(20);
        arr.add(30);
        arr.add(30);

        //size
        System.out.println(arr.size());

        //get .. A[i].. arr.get(idx)
        System.out.println(arr.get(1));

        //remove .. arr.remove(idx);
        arr.remove(1);
        System.out.println(arr.size());

```



```java
    public static void main(String[] args) {
        //Intialize
        ArrayList<Integer> arr = new ArrayList<>();
        //add
        arr.add(10);
        arr.add(20);
        arr.add(30);
        arr.add(30);

        //print
        for(int i = 0; i < arr.size(); i++){
            System.out.print(arr.get(i) + " ");
        }
        System.out.println();

        //direct print
        System.out.println(arr);

```

ArrayList with if-else

- **First Declare an ArrayList arr**.
- Then take T as an Integer input.

Format for next T Lines : $(case, x(optional))$

- **case 1**: Print the size of the ArrayList in a separate line.
- **case 2**: Print and Remove element from the last index of the ArrayList.
- **case 3**: Print x and Add x in last index of the ArrayList.
- **case 4**: Print and Remove an element from the starting (index = 0) of the ArraList.
- **case 5**: Print x and Add x at beginning (index = 0) of the ArrayList.
- **case 6**: Print all the elements from left to right that are there inside the ArrayList.

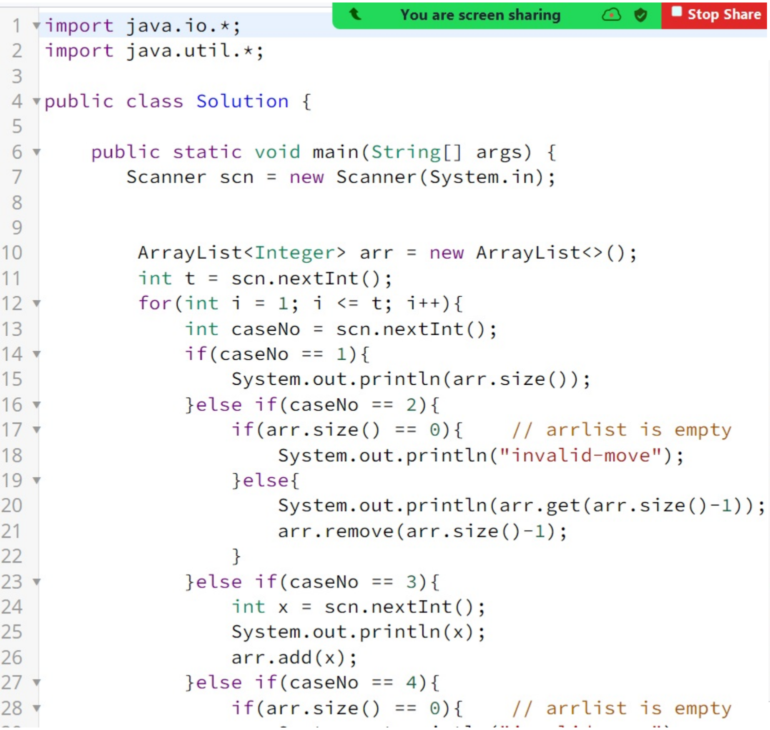
Note : In **case** $2, 4, 6$ when arr is **empty** the move is invalid, so print "**invalid-move** all lowercase".

Sample Input 0

```
8
2
6
3 2
5 1
6
1
3 3
2
```

Sample Output 0

```
invalid-move
invalid-move
2
1
1 2
2
3
3
```

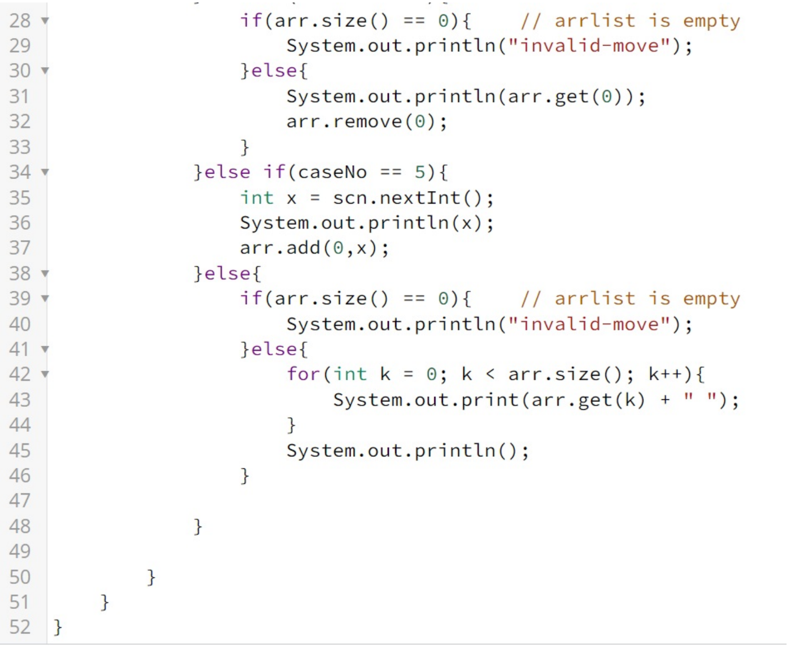


```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);


        ArrayList<Integer> arr = new ArrayList<>();
        int t = scn.nextInt();
        for(int i = 1; i <= t; i++){
            int caseNo = scn.nextInt();
            if(caseNo == 1){
                System.out.println(arr.size());
            }else if(caseNo == 2){
                if(arr.size() == 0){    // arrlist is empty
                    System.out.println("invalid-move");
                }else{
                    System.out.println(arr.get(arr.size()-1));
                    arr.remove(arr.size()-1);
                }
            }else if(caseNo == 3){
                int x = scn.nextInt();
                System.out.println(x);
                arr.add(x);
            }else if(caseNo == 4){
                if(arr.size() == 0){    // arrlist is empty
```



```java
                if(arr.size() == 0){    // arrlist is empty
                    System.out.println("invalid-move");
                }else{
                    System.out.println(arr.get(0));
                    arr.remove(0);
                }
            }else if(caseNo == 5){
                int x = scn.nextInt();
                System.out.println(x);
                arr.add(0,x);
            }else{
                if(arr.size() == 0){    // arrlist is empty
                    System.out.println("invalid-move");
                }else{
                    for(int k = 0; k < arr.size(); k++){
                        System.out.print(arr.get(k) + " ");
                    }
                    System.out.println();
                }

            }

        }
    }
}
```