

Revision.

A.L → Dynamic Array.

↳ don't define size

< Integer > Integer, ~~string~~ Integer.

• add(10)

• get(idx).

↳ valid index

• size()

• remove() i

• add(idx, 80)

AL with if else.

T=8

13

- 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 ArrayList.
- **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.

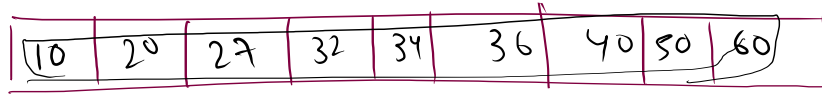
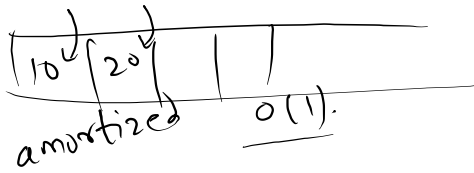
Note : In case 2, 4, 6 when arr is **empty** the move is invalid, so print "**invalid-move**" all lowercase.

8
2✓
6✓
3 2✓
5 1✓
6
1
2 3✓
2

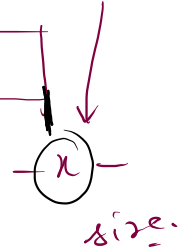
Sample Output 0

invalid-move✓
invalid-move✓
2✓
1✓
1 2
2✓
3✓
3✓

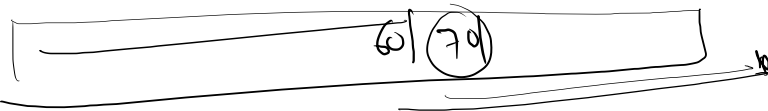
$O(1)$



initially.



string



$-2x-$

$2x \times 2$

- add(10)
- add(20)
- (27)
- (32)
- (34)

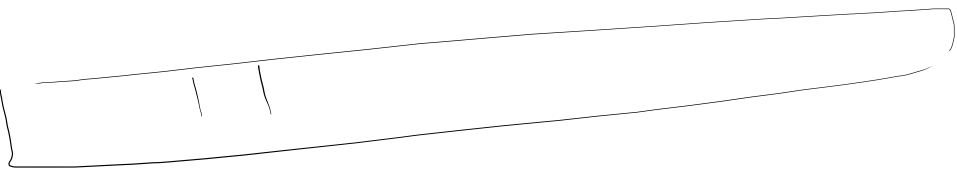
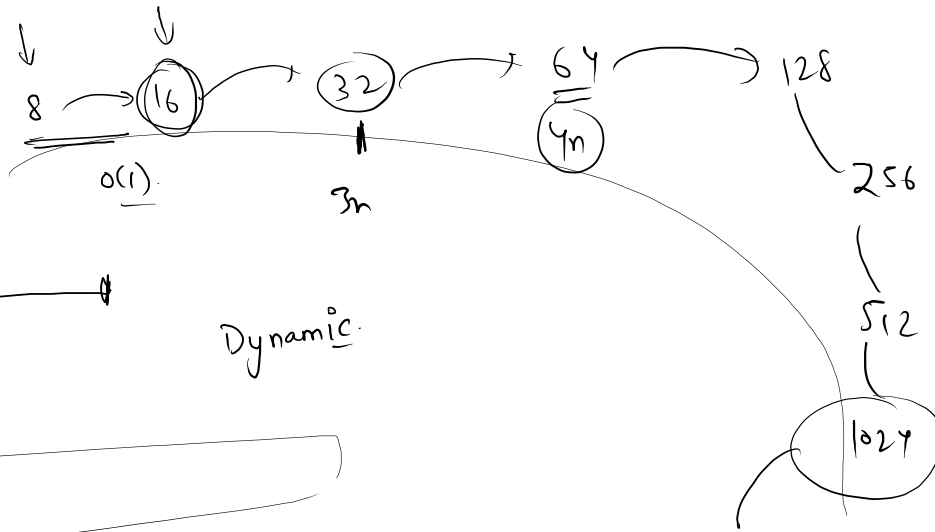


36 (70)
40
50 60

↑
pointer



⑦



```

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         ArrayList<Integer> arr = new ArrayList<>();
9         int t = scn.nextInt();
10        for(int i = 0; i < t; i++){
11            int code = scn.nextInt();
12            if(code == 1){
13                System.out.println(arr.size());
14            }
15            else if(code == 2){
16                if(arr.size() != 0){
17                    //there are some elements (non empty)
18                    int rem = arr.remove(arr.size()-1);
19                    System.out.println(rem);
20                }
21                else{
22                    System.out.println("invalid-move");
23                }
24            }
25        }
26    }
27 }

```

```

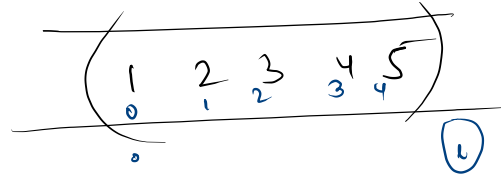
28 else if(code == 3){
29     int x = scn.nextInt();
30     System.out.println(x);
31     arr.add(x);
32 }
33 else if(code == 4){
34     if(arr.size() != 0){
35         //there are some elements (non empty)
36         int rem = arr.remove(0);
37         System.out.println(rem);
38     }
39     else{
40         System.out.println("invalid-move");
41     }
42 }
43 else if(code == 5){
44     int x = scn.nextInt();
45     System.out.println(x);
46     arr.add(0,x);
47 }else{
48     if(arr.size() != 0){
49         for(int k = 0; k < arr.size(); k++){
50             System.out.print(arr.get(k) + " ");
51         }
52         System.out.println();
53     }
54     else{
55         System.out.println("invalid-move");
56     }
57 }
58 }
59 }
60 }

```

ArrayList Printing

⑤

1 2 3 4 5



10

traditional for loop

→

1 2 3 4 5

20

for-each.

for (Integer ele : arr)

{
 pr

1 2 3 4 5

```
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
9         int n = scn.nextInt();
10
11         ArrayList<Integer> arr = new ArrayList<>();
12         for(int i = 0; i < n; i++){
13             arr.add(scn.nextInt());
14         }
15
16         for(int i = 0; i < n; i++){
17             System.out.print(arr.get(i)+ " ");
18         }
19         System.out.println();
20         for(Integer ele : arr){
21             System.out.print(ele + " ");
22         }
23     }
24 }
```

Reverse AL.

8 4 3 2 1
↓

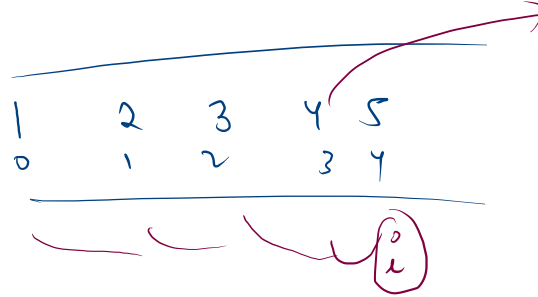
Q

1 2 3 4 5

Sample Output 0

5 4 3 2 1

5 4 3 2 1



traditional.

for (i = size - 1; i > 0; i--)

for each.
↳ reverse AL.
↳ print for each.


```
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());
    }

    //for loop reverse
    for(int i = arr.size()-1; i >= 0; i--){
        System.out.print(arr.get(i)+ " ");
    }
    System.out.println();

    //for each loop: reverse and print
    Collections.reverse(arr);
    for(int ele : arr){
        System.out.print(ele+ " ");
    }
}
```

Merge two sorted arrays 7

4
1 3 3 7
4
2 4 4 8

~~A~~ →

1
0

~~3~~

3
1

~~3~~

3
2

~~7~~

7
3

~~1~~

B →

2

4

4

8

~~8~~

~~8~~

8

```

if (A[i] > B[j])
{
    arr.add(B[j]);
    j++;
}
else
{
    arr.add(A[i]);
    i++;
}
    
```

arr



$B[j] \neq arr.get(i)$