The painter

Note :- any i^{th} index can be painted by 1 painter only

$n = 4$

arr

0	1	2	3
10	10	10	10

painter = 2

range

$si = \max(arr)$

$ei = \text{sum}(arr)$

ex :-

↓	↓	↓	↓
10	20	30	40

$p = 4$

time = 40

ex :-

↓	↓	↓	↓
10	20	30	40

$p = 1$

time = 100

time

10 . . . - - - - - - - - - - - - - - - 40

psudo code

```
si = max(arr), ei = sum(arr)
while ( si <= ei ) {
        int mid = (si+ei)/2;   // time
        if (  check( ) > K ) {
                si = mid+1;
        } else {
                ei = mid-1;
}       }
```

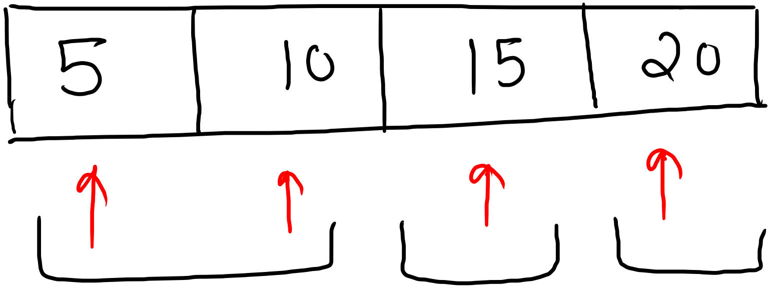
count no. of painters required

check()

random example

mid = time = 25 Yes

si = mid+1



arr	5	10	15	20

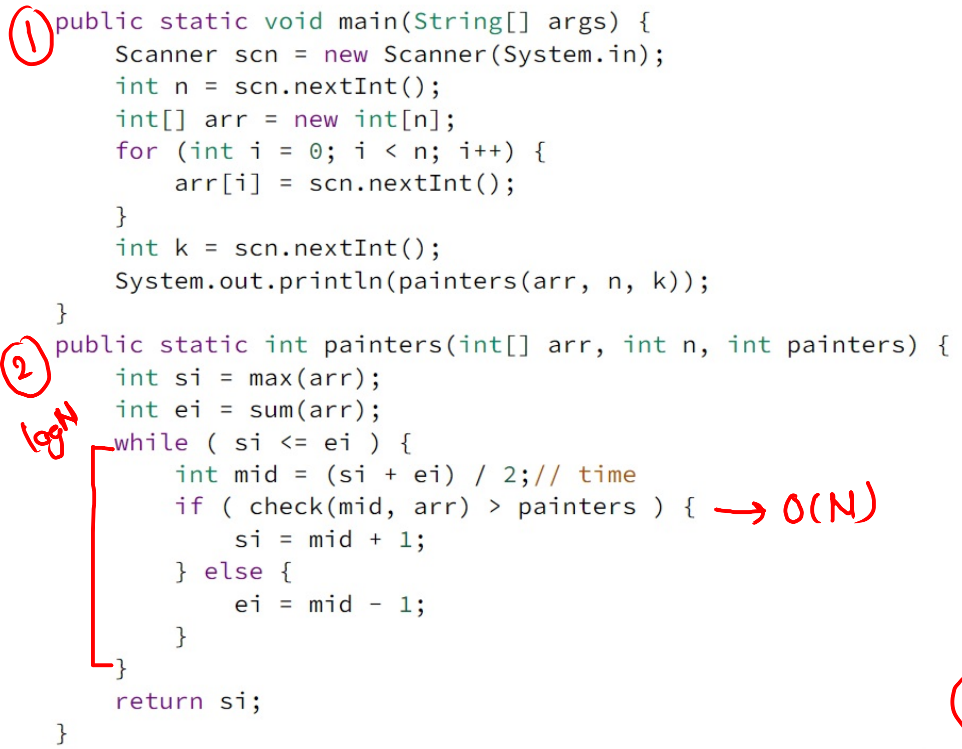
, p = 2

painter count = ~~1~~ ~~2~~ 3

→ painter1 = 5+10 < 25

→ painter2 = 15 < 25

→ painter3 = 20 < 25



1

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int k = scn.nextInt();
    System.out.println(painters(arr, n, k));
}
```

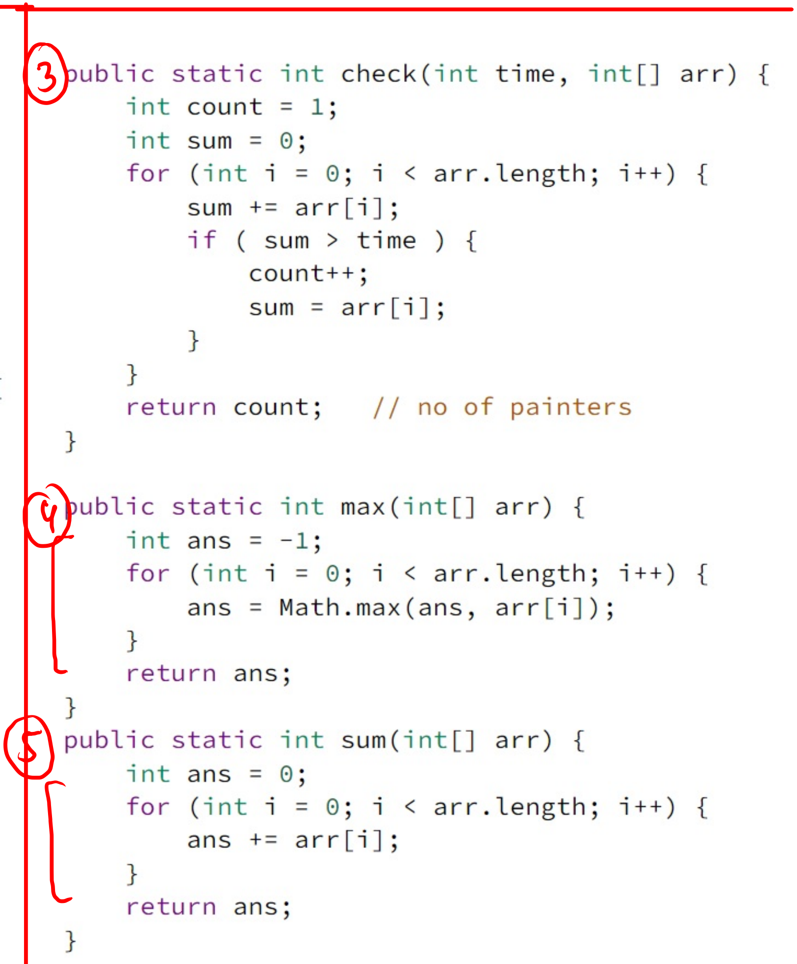
2

```java
public static int painters(int[] arr, int n, int painters) {
    int si = max(arr);
    int ei = sum(arr);
    while ( si <= ei ) {
        int mid = (si + ei) / 2;// time
        if ( check(mid, arr) > painters ) {
            si = mid + 1;
        } else {
            ei = mid - 1;
        }
    }
    return si;
}
```

logN

→ O(N)

T.C = O(N*logN)



3

```java
public static int check(int time, int[] arr) {
    int count = 1;
    int sum = 0;
    for (int i = 0; i < arr.length; i++) {
        sum += arr[i];
        if ( sum > time ) {
            count++;
            sum = arr[i];
        }
    }
    return count;    // no of painters
}
```

4

```java
public static int max(int[] arr) {
    int ans = -1;
    for (int i = 0; i < arr.length; i++) {
        ans = Math.max(ans, arr[i]);
    }
    return ans;
}
```

5

```java
public static int sum(int[] arr) {
    int ans = 0;
    for (int i = 0; i < arr.length; i++) {
        ans += arr[i];
    }
    return ans;
}
```

where, N = sum(arr) - max(arr)

⇒ Arraylist

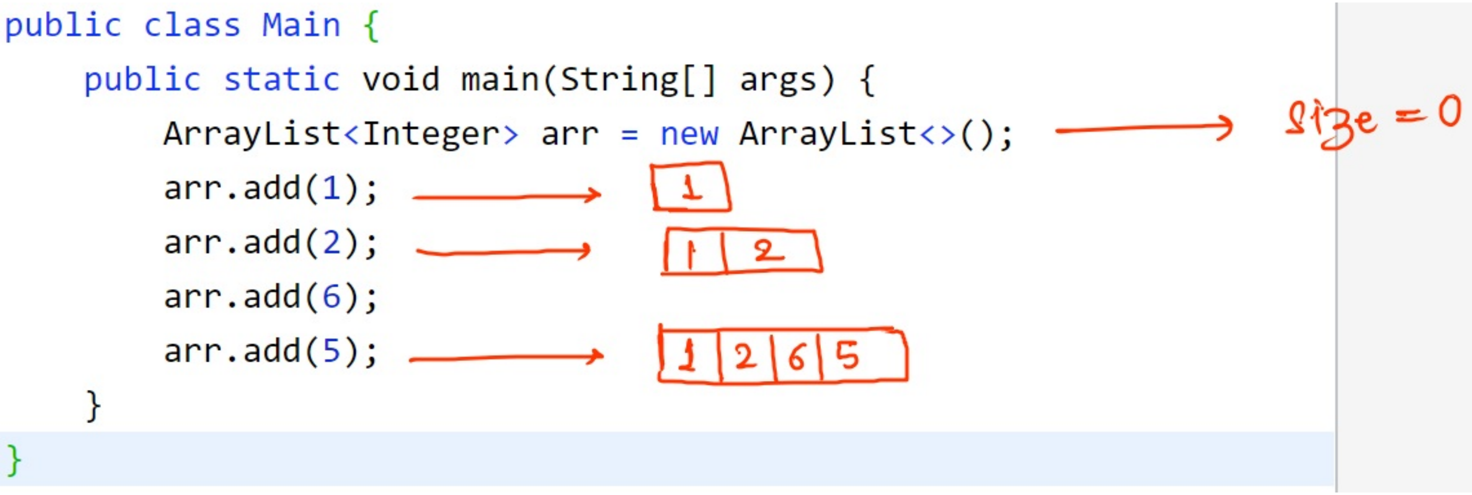
↳ dynamic array

Syntex :-

```
Arraylist<DataType> arr = new Arraylist<>();
```

↳ Integer, String, Boolean, Double - - - -

add



```java
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arr = new ArrayList<>();
        arr.add(1);
        arr.add(2);
        arr.add(6);
        arr.add(5);
    }
}
```

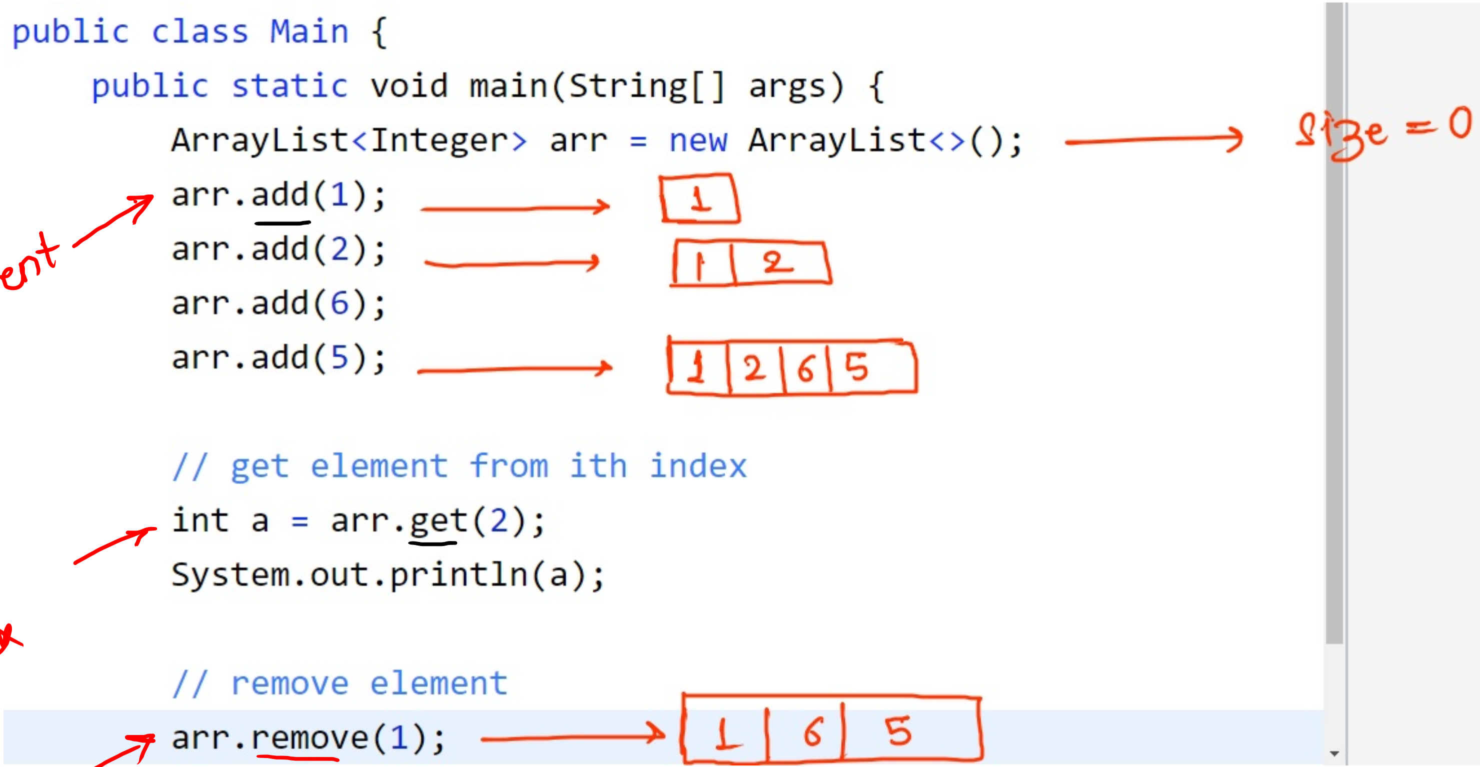
size = 0

1

1	2

1	2	6	5

↳ add, get, remove function



```
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arr = new ArrayList<>();
        arr.add(1);
        arr.add(2);
        arr.add(6);
        arr.add(5);

        // get element from ith index
        int a = arr.get(2);
        System.out.println(a);

        // remove element
        arr.remove(1);
```

}

arr.size() → 3

To sort in arraylist

↳ `Collections.sort(arr);` ↑ing order

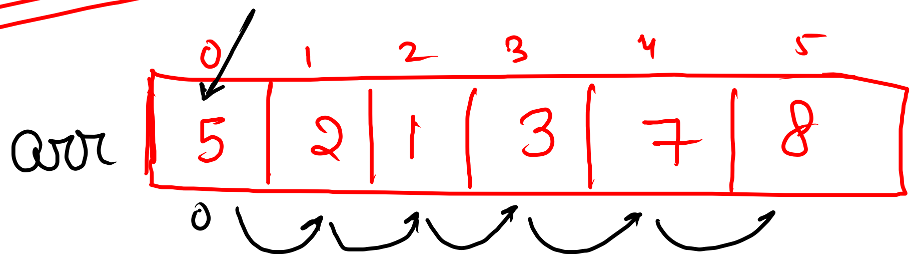
↳ `Collections.sort(arr, Collections.reverseOrder());`

To reverse the arraylist

↳ `Collections.reverse(arr);`

For each loop

i = val = arr[i]



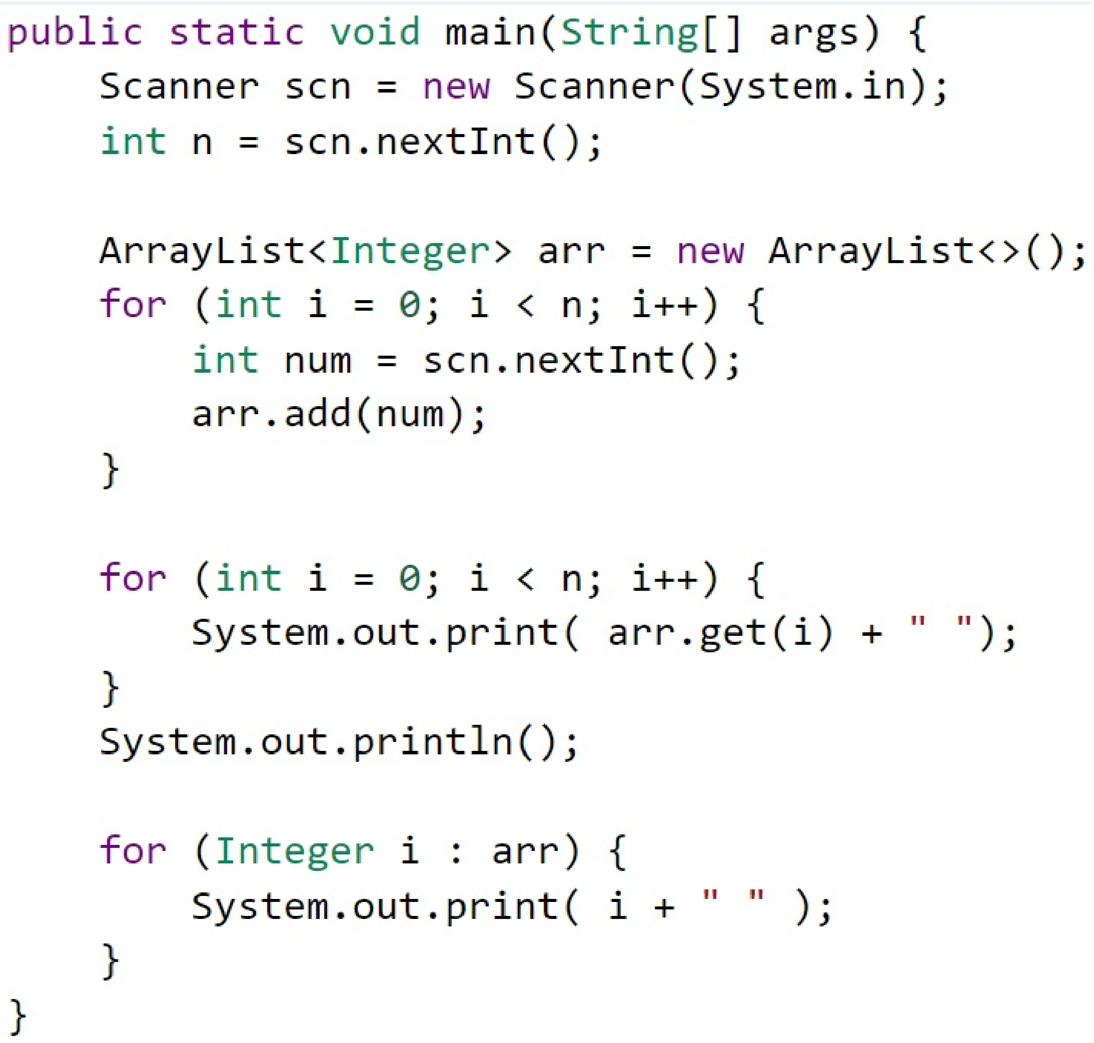
```
for ( int i : arr) {
    syso( i);
}
```

i → value

Note:- ↳ always start from 0 index and end at last idx

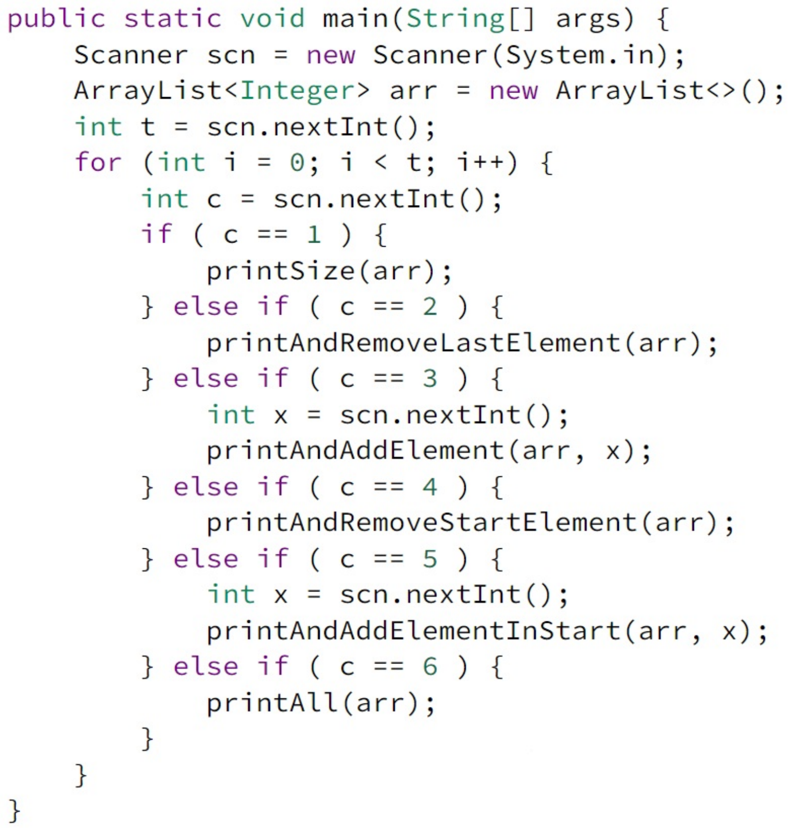
↳ can only jump by 1

ArrayList Printing



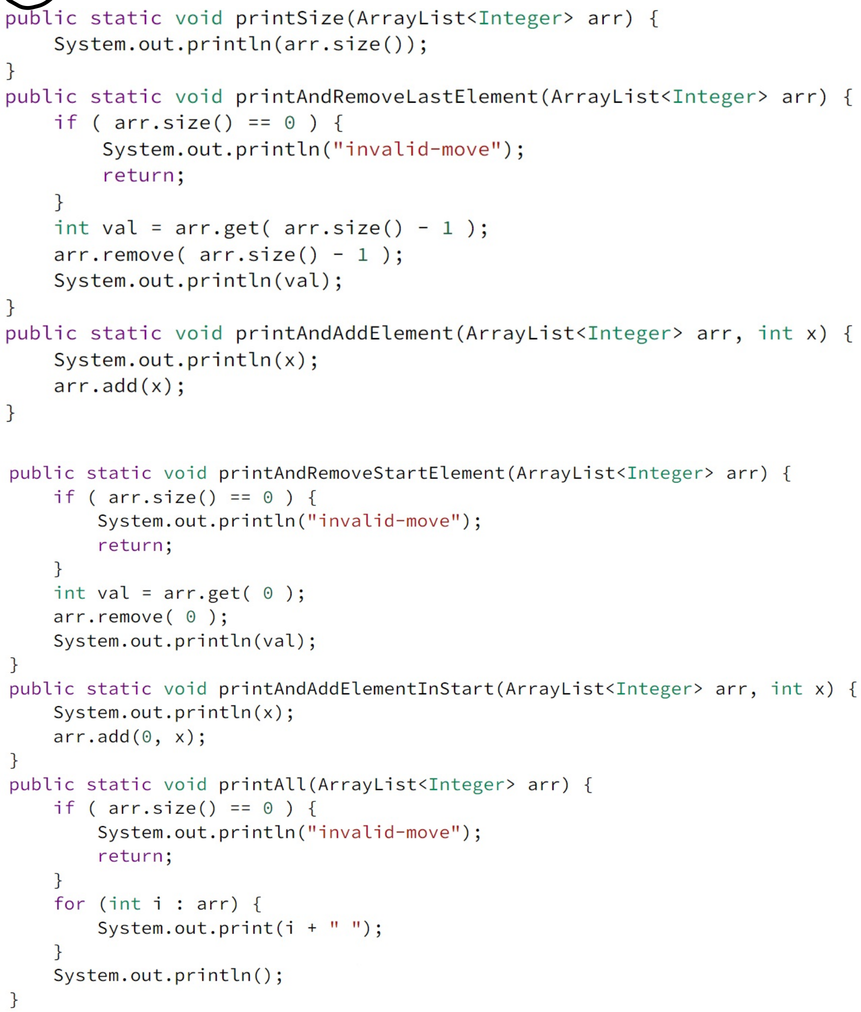
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    ArrayList<Integer> arr = new ArrayList<>();
    for (int i = 0; i < n; i++) {
        int num = scn.nextInt();
        arr.add(num);
    }

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

    for (Integer i : arr) {
        System.out.print( i + " " );
    }
}
```

1



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    ArrayList<Integer> arr = new ArrayList<>();
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int c = scn.nextInt();
        if ( c == 1 ) {
            printSize(arr);
        } else if ( c == 2 ) {
            printAndRemoveLastElement(arr);
        } else if ( c == 3 ) {
            int x = scn.nextInt();
            printAndAddElement(arr, x);
        } else if ( c == 4 ) {
            printAndRemoveStartElement(arr);
        } else if ( c == 5 ) {
            int x = scn.nextInt();
            printAndAddElementInStart(arr, x);
        } else if ( c == 6 ) {
            printAll(arr);
        }
    }
}
```

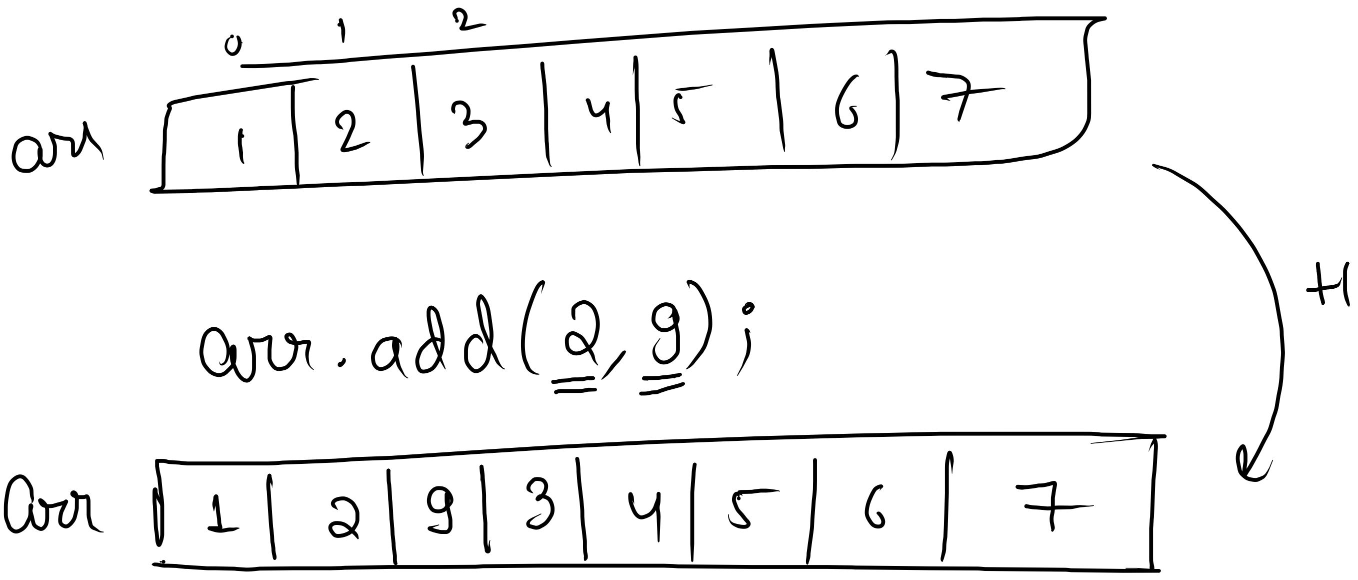
2



```java
public static void printSize(ArrayList<Integer> arr) {
    System.out.println(arr.size());
}
public static void printAndRemoveLastElement(ArrayList<Integer> arr) {
    if ( arr.size() == 0 ) {
        System.out.println("invalid-move");
        return;
    }
    int val = arr.get( arr.size() - 1 );
    arr.remove( arr.size() - 1 );
    System.out.println(val);
}
public static void printAndAddElement(ArrayList<Integer> arr, int x) {
    System.out.println(x);
    arr.add(x);
}


public static void printAndRemoveStartElement(ArrayList<Integer> arr) {
    if ( arr.size() == 0 ) {
        System.out.println("invalid-move");
        return;
    }
    int val = arr.get( 0 );
    arr.remove( 0 );
    System.out.println(val);
}
public static void printAndAddElementInStart(ArrayList<Integer> arr, int x) {
    System.out.println(x);
    arr.add(0, x);
}
public static void printAll(ArrayList<Integer> arr) {
    if ( arr.size() == 0 ) {
        System.out.println("invalid-move");
        return;
    }
    for (int i : arr) {
        System.out.print(i + " ");
    }
    System.out.println();
}
```

↳ add function

arr.add(index, val);



	0	1	2				
arr	1	2	3	4	5	6	7

+1

arr.add(2,9);

Arr	1	2	9	3	4	5	6	7