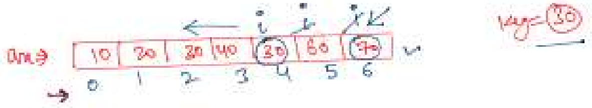
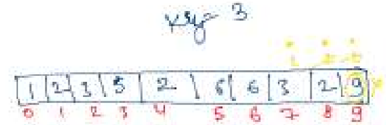
Q

You have given an array of integers of length n and a **key**, you need to find the **last index** of the key in the given array . If not present, then return -1.



i
return;



ans = -1; 7

7

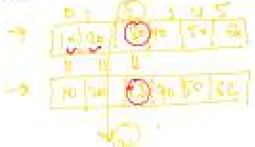
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];

    for(int i=0;i<n;i++){
        arr[i]= scn.nextInt();
    }
    int key = scn.nextInt();
    int ans =-1;
    for(int i = arr.length-1;i>=0;i--){
        if(arr[i]==key){
            ans =i;
            break;
        }
    }
    System.out.println(ans);
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Yo
}
```

Q

Declare the first array of size **n** that stores values of int data-type. Then take **n** integer inputs and store them in the array one by one.

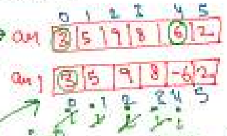
Then again declare a **second array of size n** that stores values of int data-type. Then take n integer inputs and store them in the array one by one. Then print the **index** at which you find the first non matching number in the array.

Input Format



2

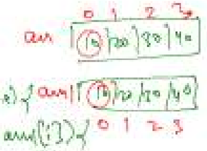
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];

    for(int i=0;i<n;i++){
        arr[i]= scn.nextInt();
    }
    int[] arr1 = new int[n];

    for(int i=0;i<n;i++){
        arr1[i]= scn.nextInt();
    }

    for(int i =0;i<arr.length;i++){
        if(arr[i] != arr1[i]){
            System.out.println(i);
            return;
        }
    }
    /* Enter your code here. Read input from STDIN. Print output to STDOUT
}
```



arr[i] != arr1[i]

6 != -6

4



int ans = -1;

for(int i=0; i<n; i++){

if(arr[i] != arr1[i]){

ans = i;

} break;

}

Syso(ans);

-1

ans = 0

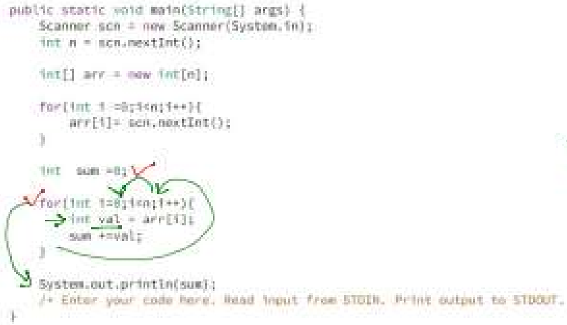
Q

arr =

5	6	9	3	2	1	-1
0	1	2	3	4	5	6

~~i~~ ~~i~~ ~~i~~ ~~i~~ ~~i~~ ~~i~~ ~~i~~ i

sum = ~~0~~ ~~5~~ ~~11~~ ~~20~~ ~~23~~ ~~25~~ ~~26~~ 25

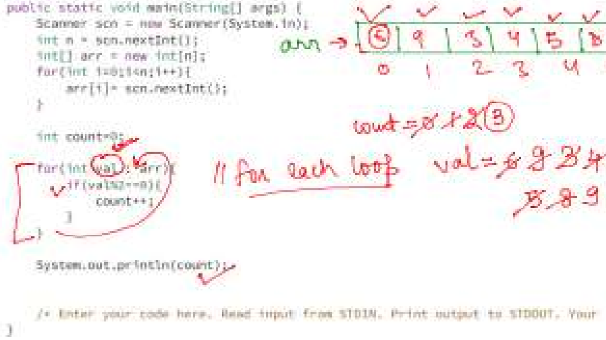


```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int[] arr = new int[n];

    for(int i =0;i<n;i++){
        arr[i]= scn.nextInt();
    }

    int  sum =0;

    for(int i=0;i<n;i++){
        int val = arr[i];
        sum +=val;
    }

    System.out.println(sum);
    /* Enter your code here. Read input from STDIN. Print output to STDOUT.
}
```

25

count even

Q



```
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 count=0;

    for(int val : arr){
        if(val%2==0){
            count++;
        }
    }

    System.out.println(count);

    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your
}
```

arr →

6	9	3	4	5	8	9
0	1	2	3	4	5	6

count = ~~0~~ ~~1~~ ~~2~~ 3

// for each loop

val = ~~6~~ ~~9~~ ~~3~~ ~~4~~ ~~5~~ ~~8~~ 9

① Traverse whole array starting from '0' index.

for (int val : arr){
Syso (val);

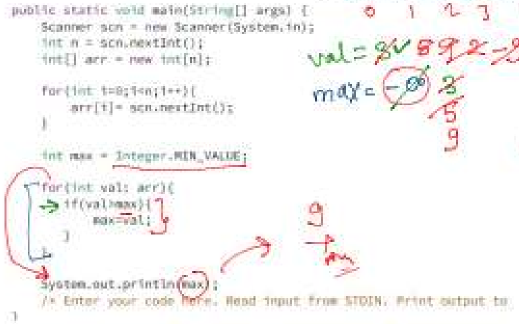
foreach

arr[i]
0 to n-1

x

arr →

3	5	9	2	-9	8	4
0	1	2	3	4	5	6



```
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 max = Integer.MIN_VALUE;

    for(int val: arr){
        if(val>max){
            max=val;
        }
    }

    System.out.println(max);
    /* Enter your code here. Read input from STDIN. Print output to
}
```

val = ~~3~~ ~~5~~ ~~9~~ ~~2~~ ~~-9~~ ~~8~~ 6

max = ~~-∞~~ ~~3~~ ~~5~~ 9

9

max = -∞

min = +∞

Integer.MAX_VALUE

max = ~~-∞~~ ~~-2~~ -1

-2	-4	-8	-1

Q

→ | 1 | 2 | 3 | 0 | 5 |

i

error

1st logic   product = 0

2 Test

0   0/1   0/2   0/3   0/0   0/5
      ↓   ↓   ↓   ⇓   ⇓
      0   0   0   error   0.

j →

2nd ⇒ | 1 | 2 | 3 | 0 | 5 |

→ i   i   i
0   1   2   3   4

n x n = O(n²)

= O(n)

① ~~Time~~

```
i=0
for(int i=0; i<n; i++){
    int prod=1;
    for(int j=0; i<n; j++){
        if (i!=j){
            prod *=arr[i];
        } else { continue;
        }
    } syso(prod);
}
```

arr → | 1 | 2 | 3 | 0 | 5 |

count = 1

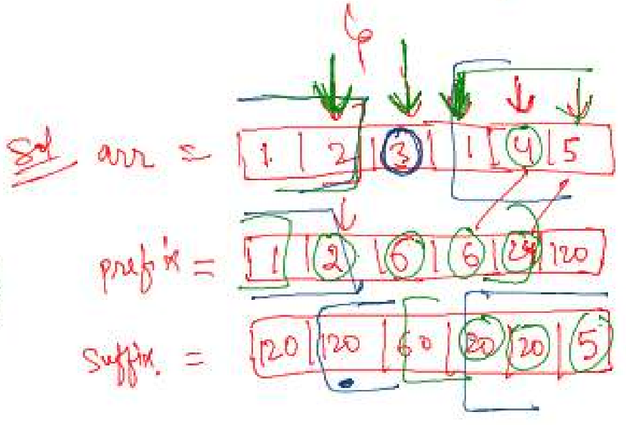
product = 30

```
if (arr[i]==0){
    count++;
} else{
    prod = prod * arr[i];
}
```

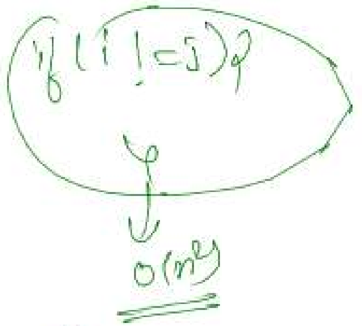
Solution

```
for( i=0 to arr.length ){
    if (count > 1){
        syso("0");
    }
            && arr[i] != 0){
```

```
else if( cou
            Syso("0");
        }
    else if( count==1 && arr[i]==0){
            Syso( prod);
        }

        else{  Syso( prod/arr[i]);

        }
}
```



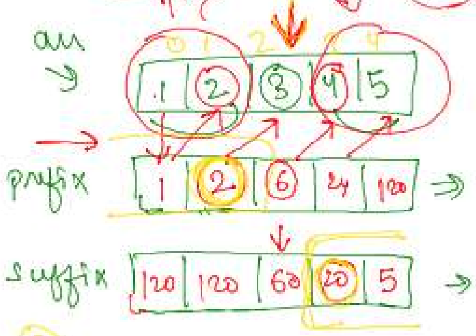
Sol arr = 1 | 2 | 3 | 1 | 4 | 5
prefix = 1 | 2 | 6 | 6 | 24 | 120
suffix = 120 | 120 | 60 | 20 | 20 | 5
O(n)



i
pref[i-1] * suff[i+1]

→ 120 | 60 | 40 | 120 | 30 | 24

2 → 20 = 40



prefix ⇒ to store product of all elements before me

suffix → to store product of all element after me

i = 2

prefix[i-1] * suf[i+1]

pre[1] * suu[3]

40