Revision.

Collection of similar data.

same data type

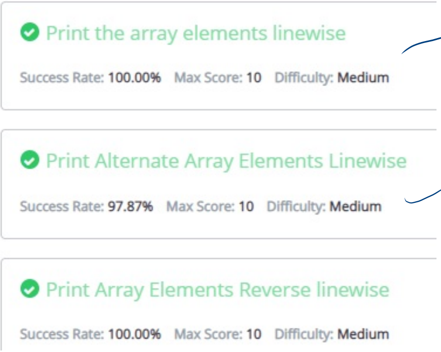
Stored continuwosly.

marks[4] = 17;

int

marks	0	0	0	0	~~0~~ 17	0
	0	1	2	3	4	5

} → index.



✓ Print the array elements linewise

Success Rate: 100.00% Max Score: 10 Difficulty: Medium

✓ Print Alternate Array Elements Linewise

Success Rate: 97.87% Max Score: 10 Difficulty: Medium

✓ Print Array Elements Reverse linewise

Success Rate: 100.00% Max Score: 10 Difficulty: Medium

for

9	3	1	5	7
0	1	2	3	4

9
3
1
5
7

```
i < n
i < A.length
Syso(A[i]);
```

②

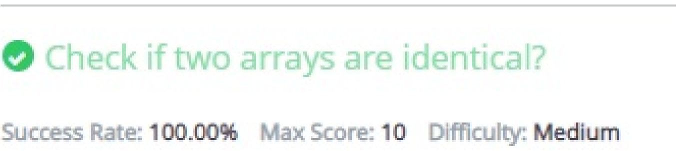
```
for( i=0 ; i+2)
{   Syso ( A[i]);
}
```

9
1
7

③

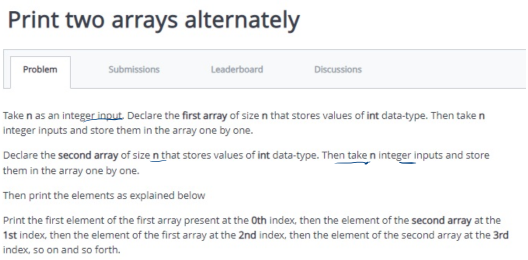
7
5
1
3
9

```
for( i=n-1; i≥0; i--
{
    A(i);
```



Check if two arrays are identical?

Success Rate: 100.00% Max Score: 10 Difficulty: Medium



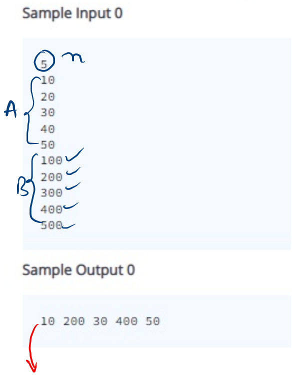
Print two arrays alternately

Take **n** as an integer input. 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.

Declare the **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 elements as explained below

Print the first element of the first array present at the **0th** index, then the element of the **second array** at the **1st** index, then the element of the first array at the **2nd** index, then the element of the second array at the **3rd** index, so on and so forth.

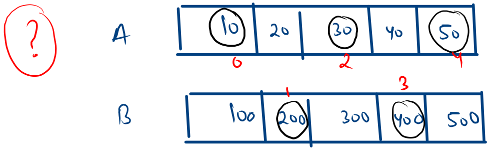


Sample Input 0

```
5
10
20
30
40
50
100
200
300
400
500
```

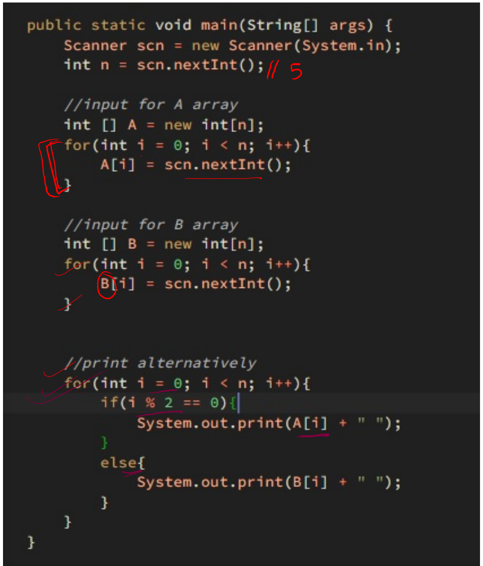
Sample Output 0

```
10 200 30 400 50
```

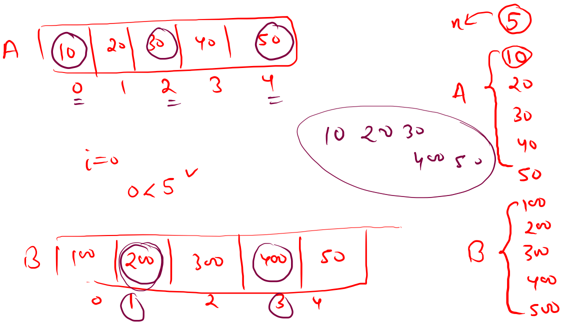


o/p

10 200 30 400 50



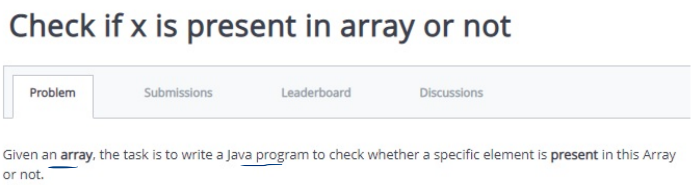
```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();// 5

    //input for A array
    int [] A = new int[n];
    for(int i = 0; i < n; i++){
        A[i] = scn.nextInt();
    }

    //input for B array
    int [] B = new int[n];
    for(int i = 0; i < n; i++){
        B[i] = scn.nextInt();
    }


    //print alternatively
    for(int i = 0; i < n; i++){
        if(i % 2 == 0){
            System.out.print(A[i] + " ");
        }
        else{
            System.out.print(B[i] + " ");
        }
    }
}
```



for (______________)

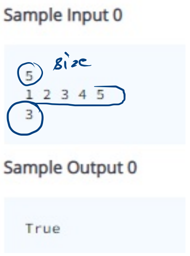
even → A

odd → B



Check if x is present in array or not

Given an array, the task is to write a Java program to check whether a specific element is present in this Array or not.



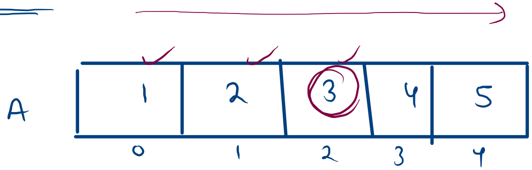
Sample Input 0

```
5   size
1 2 3 4 5
3
```

Sample Output 0

```
True
```

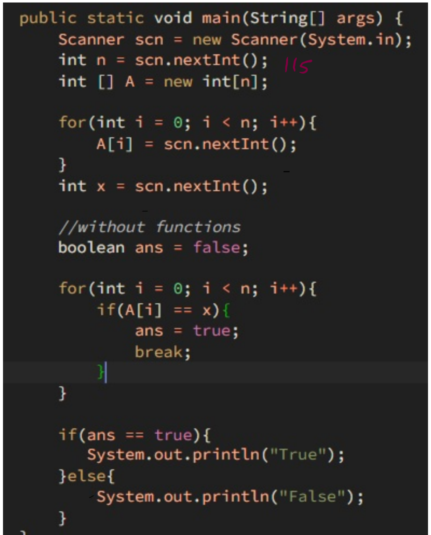
linear Search



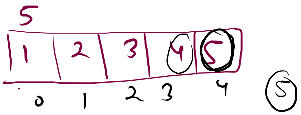
x = 3 ________ True

33. ________ False.

?



```
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();

    //without functions
    boolean ans = false;

    for(int i = 0; i < n; i++){
        if(A[i] == x){
            ans = true;
            break;
        }
    }

    if(ans == true){
       System.out.println("True");
    }else{
       System.out.println("False");
    }
```



x = 44

ans = ~~false~~ true

i = ~~0~~ ~~1~~ ~~2~~ 3

0 < 5 ✓

A[0] == x

1 == 4 X

1 < 5 ✓

A[1] == x

2 == 4

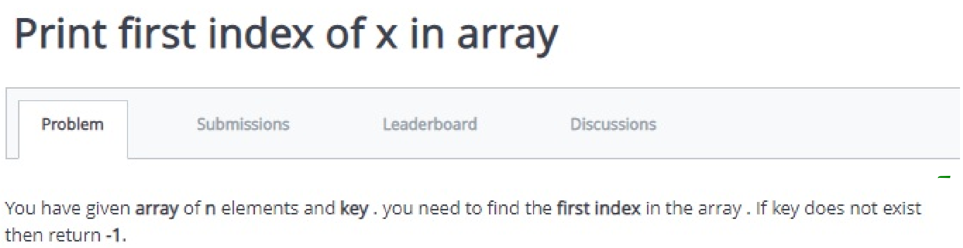
2 < 5 ✓

A[2] == 4

3 == 4

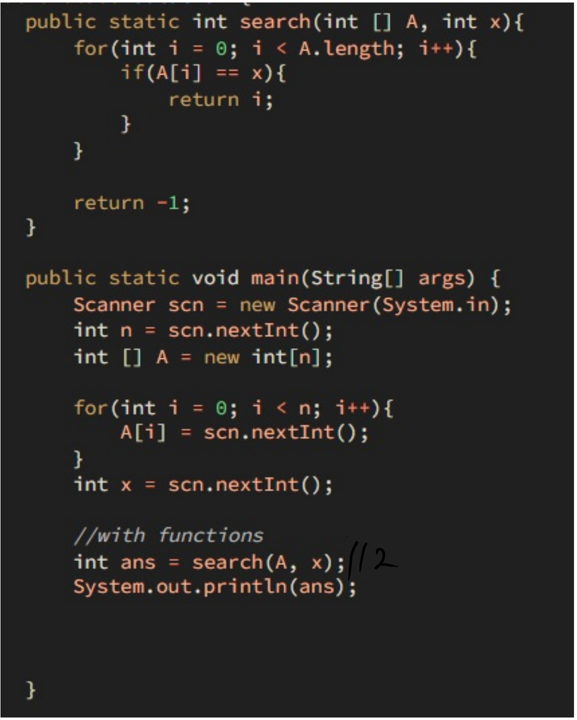
3 < 5 ✓

A[3] == 4



Print first index of x in array

You have given **array** of n elements and **key** . you need to find the **first index** in the array . If key does not exist then return **-1**.



```
public static int search(int [] A, int x){
    for(int i = 0; i < A.length; i++){
        if(A[i] == x){
            return i;
        }
    }

    return -1;
}

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();

    //with functions
    int ans = search(A, x);//2
    System.out.println(ans);



}
```

5

A →

1	2	3	3	3
0	1	2	3	4

i=0

A(0) == 3

1 == 3 ×

i=1

A(1) == 3

2 == 3

i=2

A(2) == 3.

3 == 3

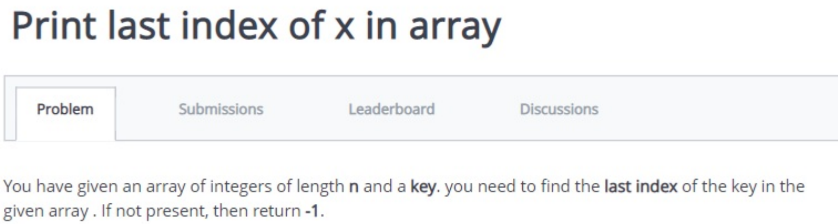
x = 3

index.

0 1 2 3 4

-1

int



Print last index of x in array

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**.

5

1	2	3	3	3	4
0	1	2	3	4	

$x = 3.$

first occurence

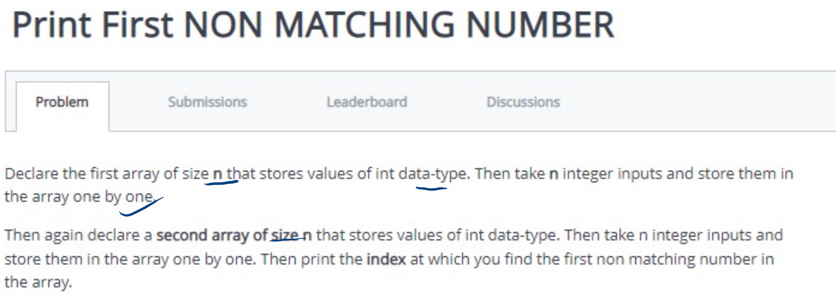
$x=3$

fo → 2

lo → 4

1	2	3	3	3	4
0	1	2	3	4	5

last occurence



Print First NON MATCHING NUMBER

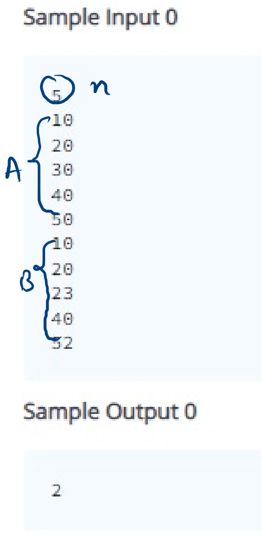
Declare the first array of size n that stores values of int data-type. Then take **n** integer inputs and store them in the array one by one.

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.

n =5

10	20	30	40	50
0	1	2	3	4

10	20	23	40	52
0	1	2	3	4



Sample Input 0

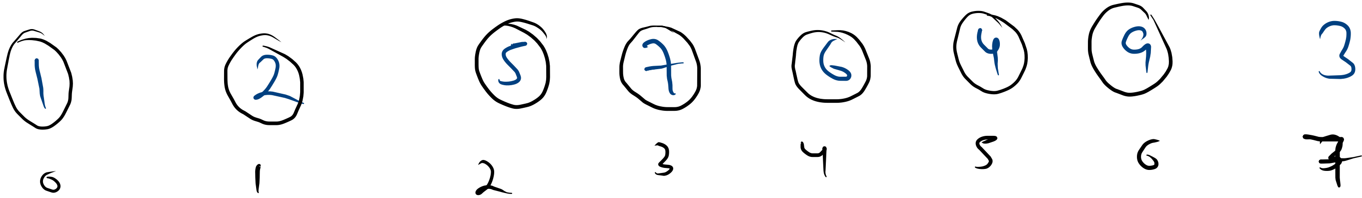
```
5
10
20
30
40
50
10
20
23
40
52
```

Sample Output 0

```
2
```

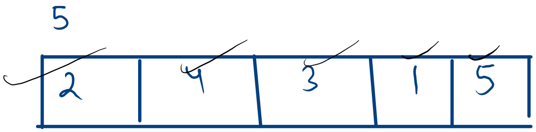
linear Search.

$x = 9$



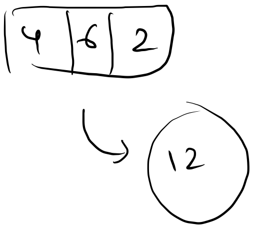
1	2	5	7	6	4	9	3
0	1	2	3	4	5	6	7

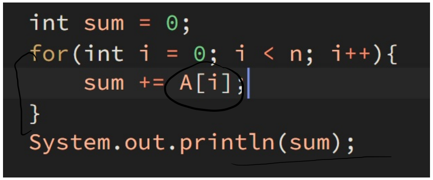
Sum of all Elements of Array



sum = 0

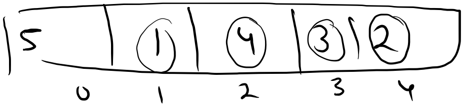
sum += A[i]





```
int sum = 0;
for(int i = 0; i < n; i++){
    sum += A[i];
}
System.out.println(sum);
```

n=5



sum = ~~0~~ ~~5~~ ~~6~~ ~~10~~ ~~13~~ 15

i = ~~0~~ ~~1~~ 0<5 ✓

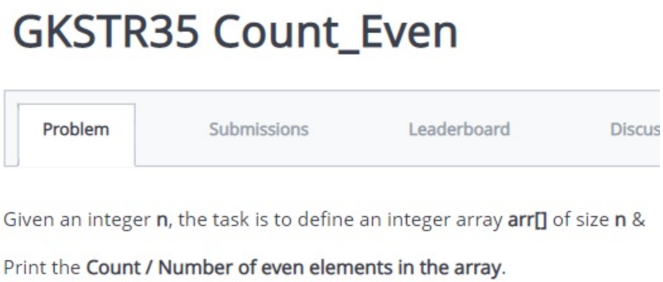
~~2~~ 1<5 ✓

2<5 ✓

3 3<5 ✓

4 4<5 ✓

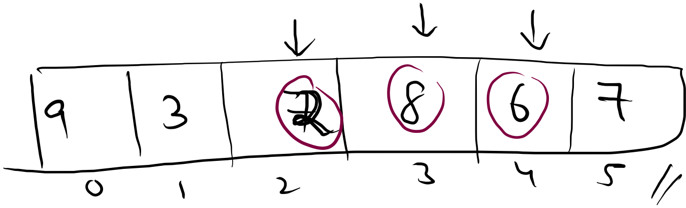
5 5<5 ✗



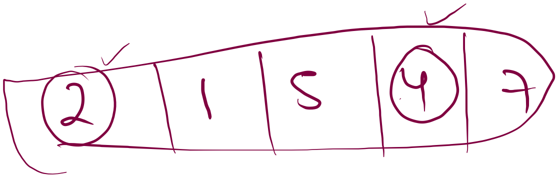
GKSTR35 Count_Even

Given an integer **n**, the task is to define an integer array **arr[]** of size **n** &

Print the **Count / Number of even elements in the array**.



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



ans = 2

ans = 3.