Print first index of x in array

n=8

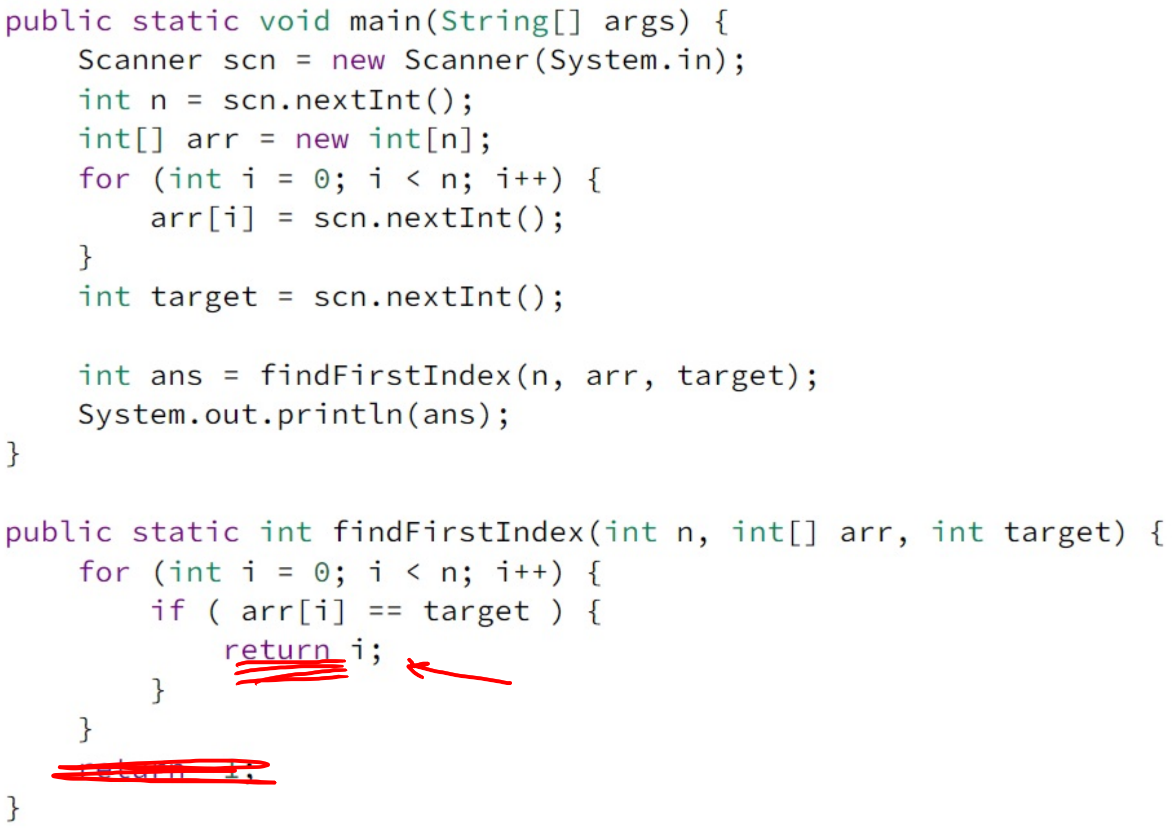
3	3	2	1	0	7	2	5
0	1	2	3	4	5	6	7

size=8

target = 2

ans = 6

raise hand if submitted

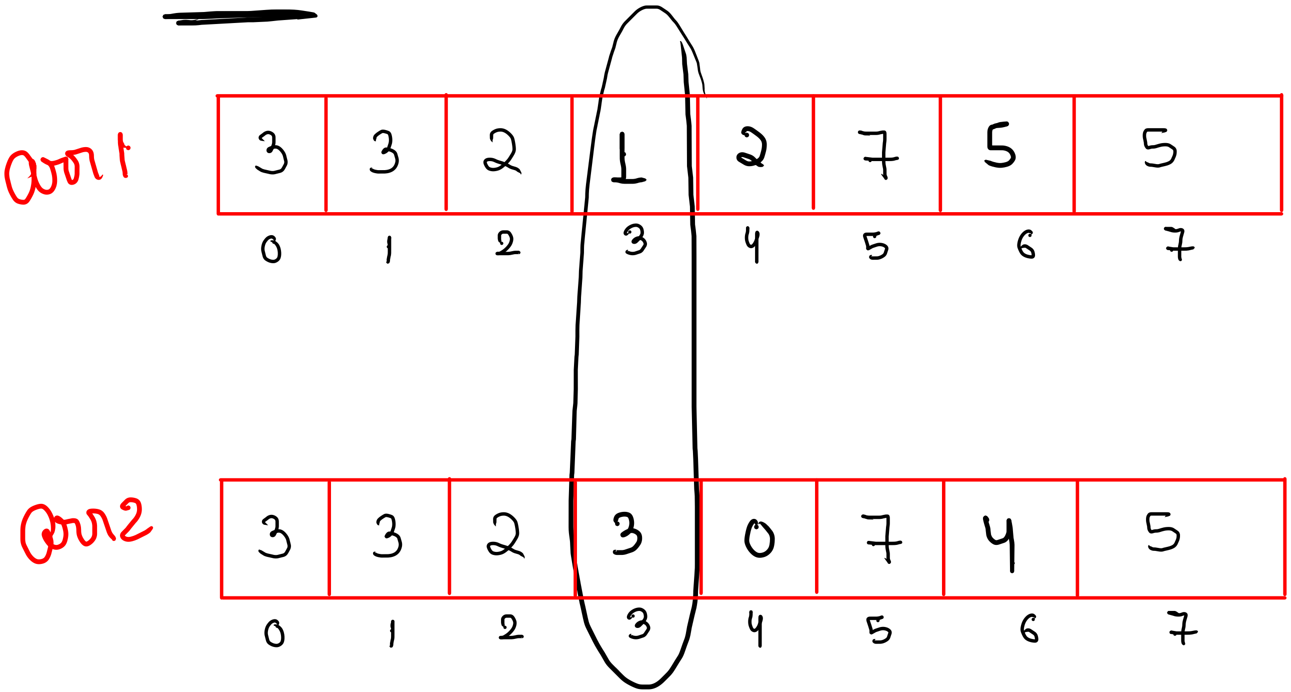


```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int target = scn.nextInt();

    int ans = findFirstIndex(n, arr, target);
    System.out.println(ans);
}

public static int findFirstIndex(int n, int[] arr, int target) {
    for (int i = 0; i < n; i++) {
        if ( arr[i] == target ) {
            return i;
        }
    }
    ~~return -1;~~
}
```

Print First NON MATCHING NUMBER



arr1	3	3	2	1	2	7	5	5
	0	1	2	3	4	5	6	7

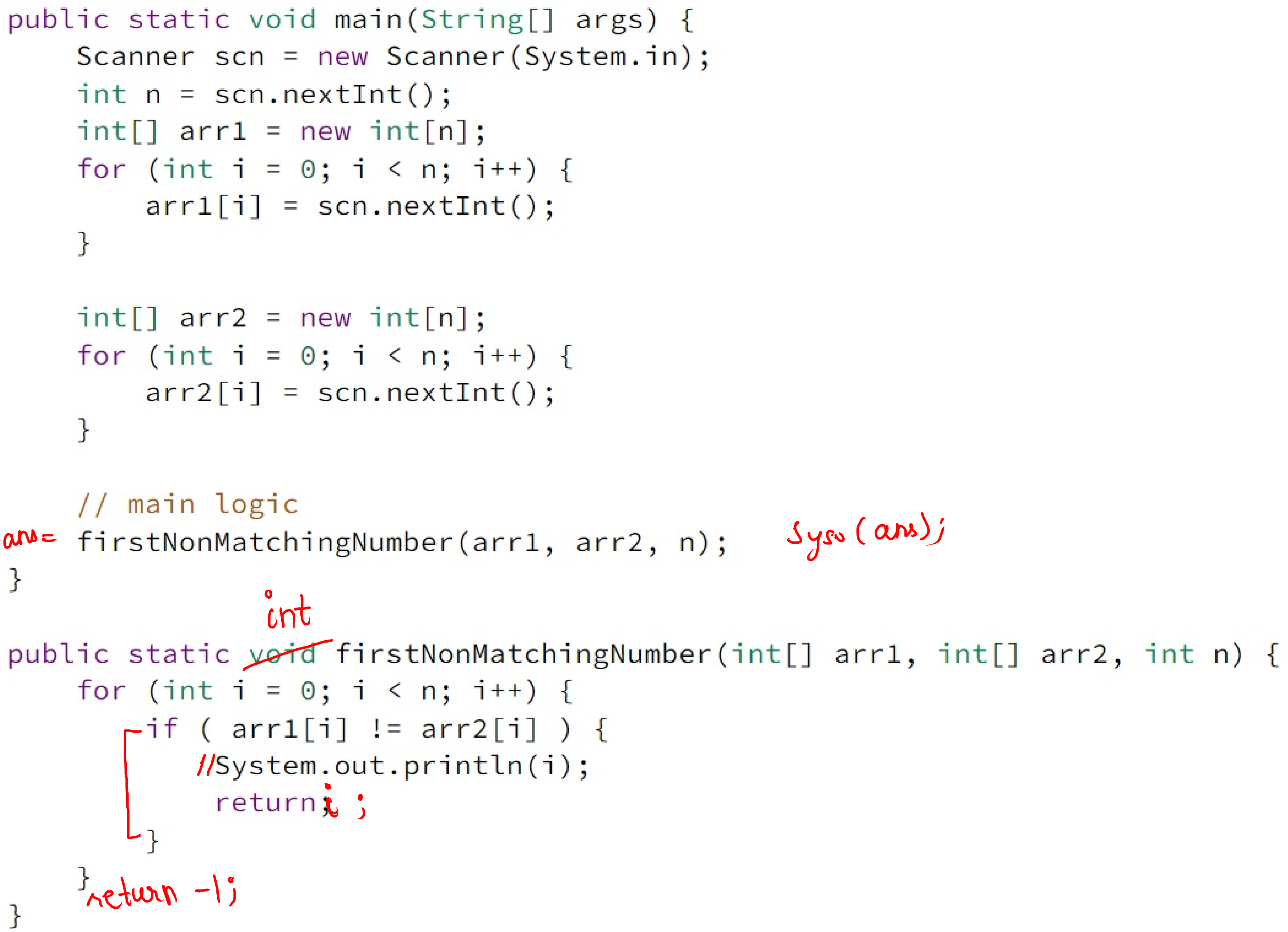
arr2	3	3	2	3	0	7	4	5
	0	1	2	3	4	5	6	7

ans = 3

logic

if (arr1[i] != arr2[i])

code



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr1 = new int[n];
    for (int i = 0; i < n; i++) {
        arr1[i] = scn.nextInt();
    }

    int[] arr2 = new int[n];
    for (int i = 0; i < n; i++) {
        arr2[i] = scn.nextInt();
    }

    // main logic
int ans= firstNonMatchingNumber(arr1, arr2, n);      Syso(ans);
}

                 int
public static ~~void~~ firstNonMatchingNumber(int[] arr1, int[] arr2, int n) {
    for (int i = 0; i < n; i++) {
        if ( arr1[i] != arr2[i] ) {
           //System.out.println(i);
            return i ;
        }
    } return -1;
}
```

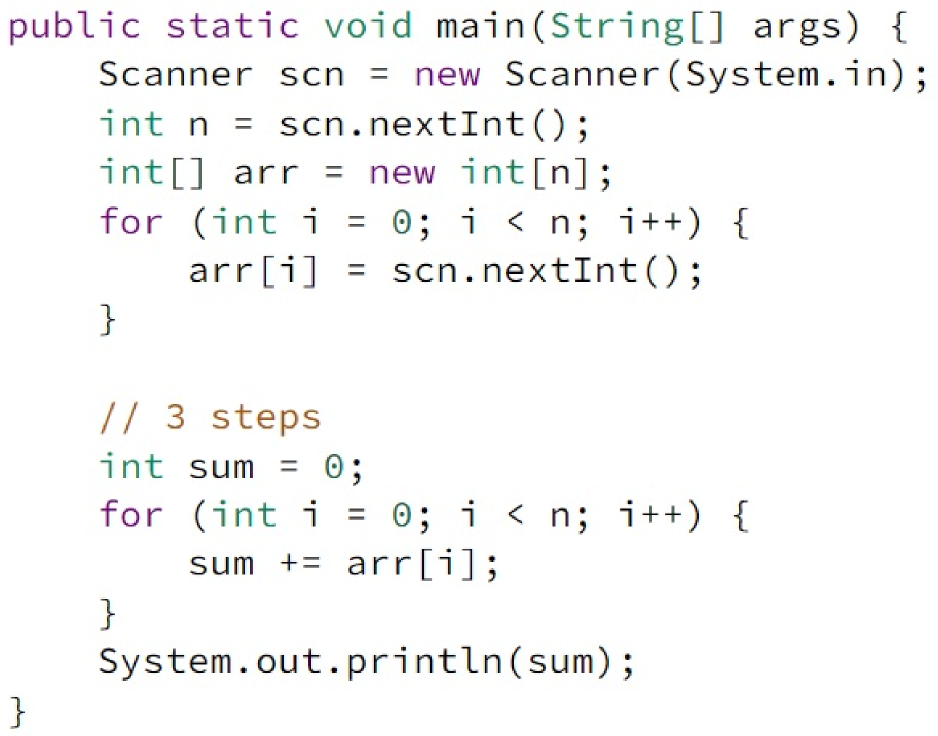
Sum of all Elements of Array

arr	5	3	1	7	-8	0	3	-1	10
	0	1	2	3	4	5	6	7	8

psudo code

```
1) declare sum with initial value zero
2) Iterate over array from start to end
    2.1) pick each element and add it in sum
3) return sum
```

code



```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();
    }

    // 3 steps
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    System.out.println(sum);
}
```

Maximum of Array

dry run

0	1	2	3	4
1	5	0	3	7

ans= 7

Integer.MAX_VALUE; ⇒ ∞

Integer.MIN_VALUE; ⇒ -∞

n=11

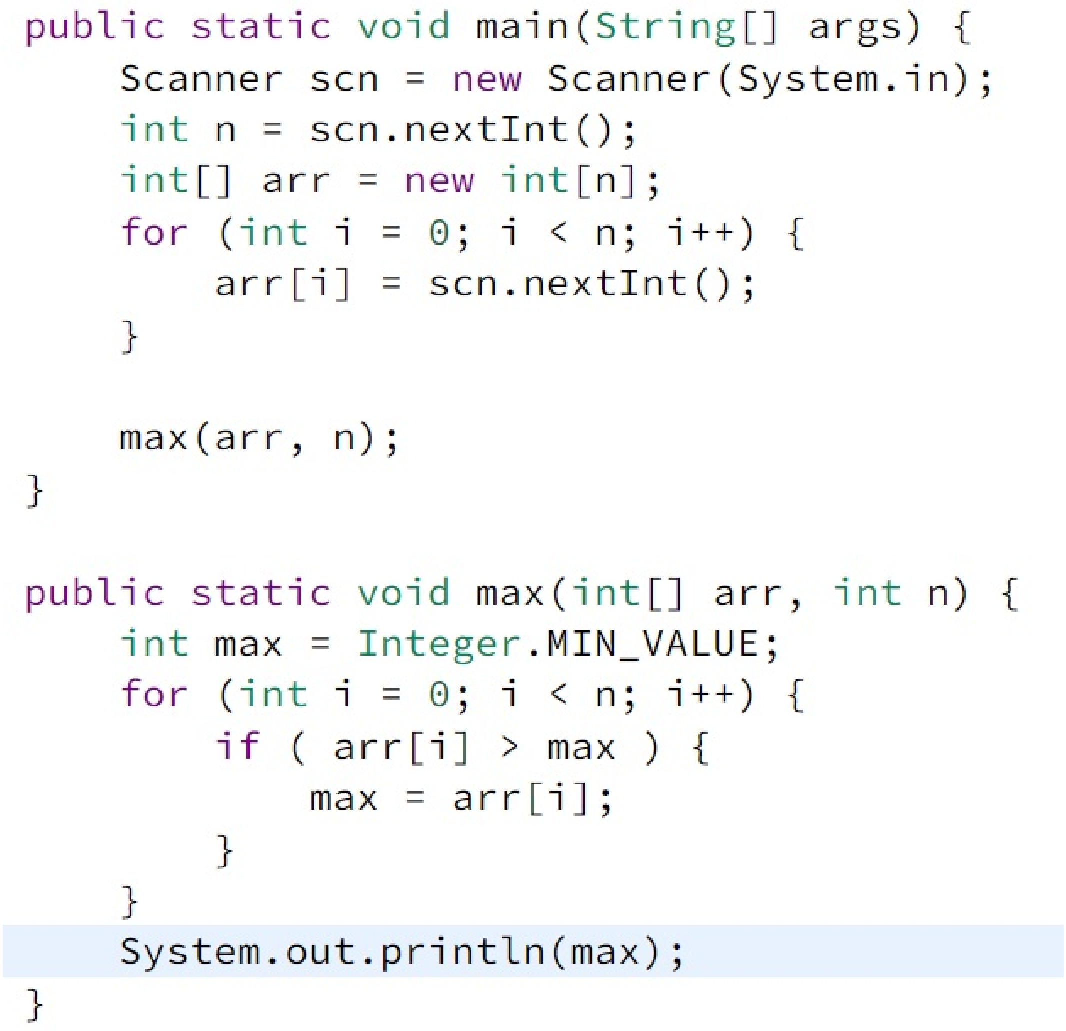
arr

2	5	3	1	0	7	3	0	8	9	11
0	1	2	3	4	5	6	7	8	9	10

```
int maxi= Integer.MIN_VALUE;
for( int i=0; i<n; i++){
    if( arr[i] > maxi){
        maxi = arr[i];
    }
}
return maxi;
```

psudo :-

```
1) declare maxi with value -∞
2) Iterate from start to end
    2.1) compare current element with maxi
    2.2) consider the larger value
3) Return maxi.
```



```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();
    }

    max(arr, n);
}

public static void max(int[] arr, int n) {
    int max = Integer.MIN_VALUE;
    for (int i = 0; i < n; i++) {
        if ( arr[i] > max ) {
            max = arr[i];
        }
    }
    System.out.println(max);
}
```

dry run

n=11

arr	2	5	3	1	0	7	3	0	8	9	11
	0	1	2	3	4	5	6	7	8	9	10

maxi = ~~-∞~~ ~~2~~ ~~5~~ ~~7~~ ~~8~~ ~~9~~ 11

i=0, 2 > -∞ ✓

i=1, 5 > 2 ✓

i=2, 3 > 5 ✗

i=3, 1 > 5 ✗

i=4, 0 > 5 ✗

i=5, 7 > 5 ✓

i=6, 3 > 7 ✗

i=7, 0 > 7 ✗

i=8, 8 > 7 ✓

i=9, 9 > 8 ✓

i=10, 11 > 9 ✓