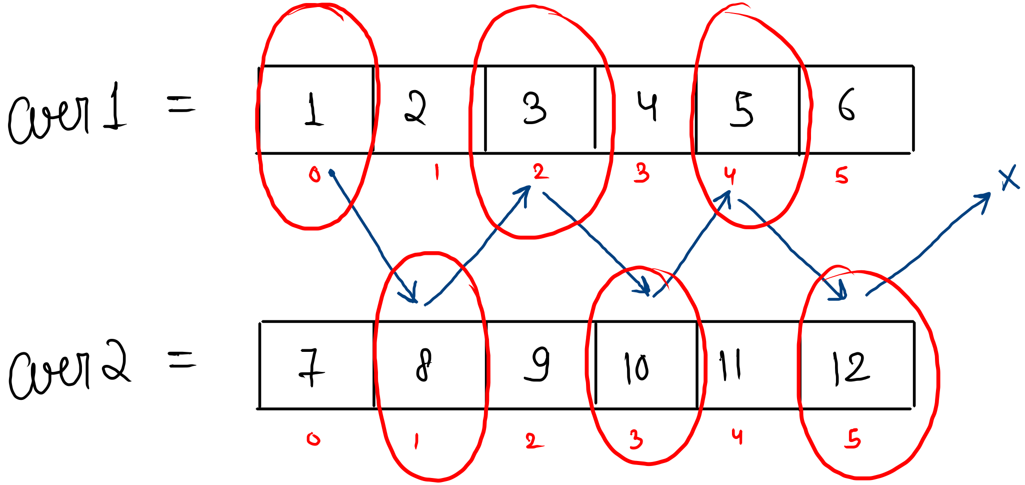
Print two arrays alternately

n= 6



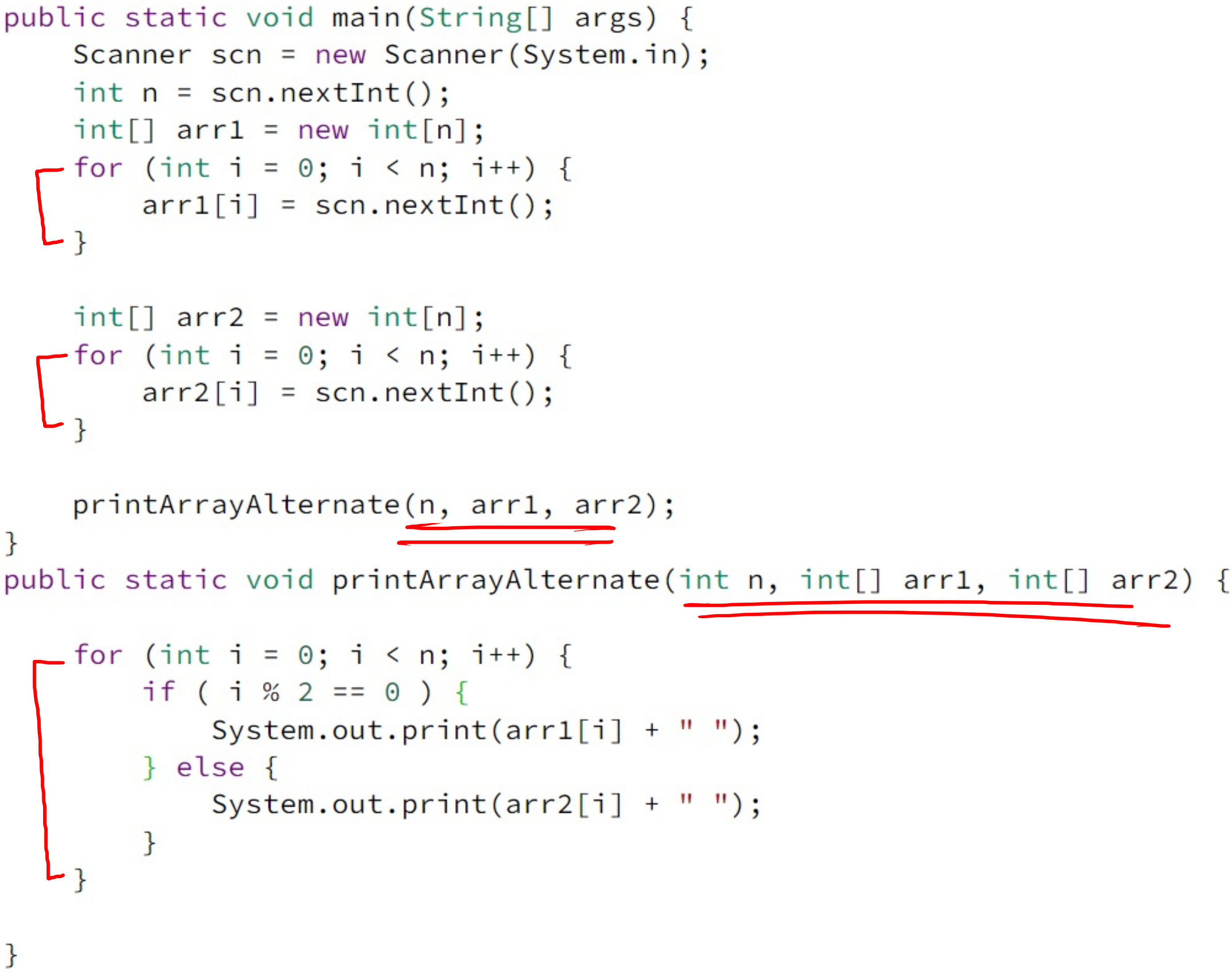
arr1 =	1	2	3	4	5	6
	0	1	2	3	4	5
arr2 =	7	8	9	10	11	12
	0	1	2	3	4	5

o/p :- 1 8 3 10 5 12

observation

index		
0	→	arr1
1	→	arr2
2	→	arr1
3	→	arr2
4	→	arr1
5	→	arr2

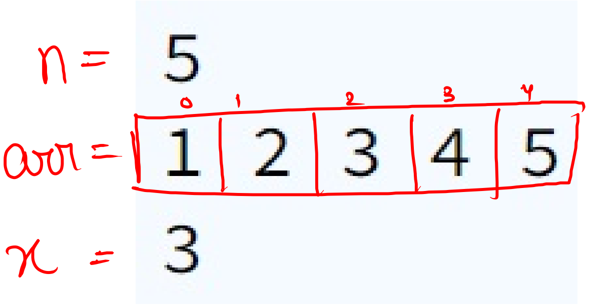
code



```java
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();
    }

    printArrayAlternate(n, arr1, arr2);
}
public static void printArrayAlternate(int n, int[] arr1, int[] arr2) {

    for (int i = 0; i < n; i++) {
        if ( i % 2 == 0 ) {
            System.out.print(arr1[i] + " ");
        } else {
            System.out.print(arr2[i] + " ");
        }
    }

}
```

⟹ Searching in array

Check if x is present in array or not



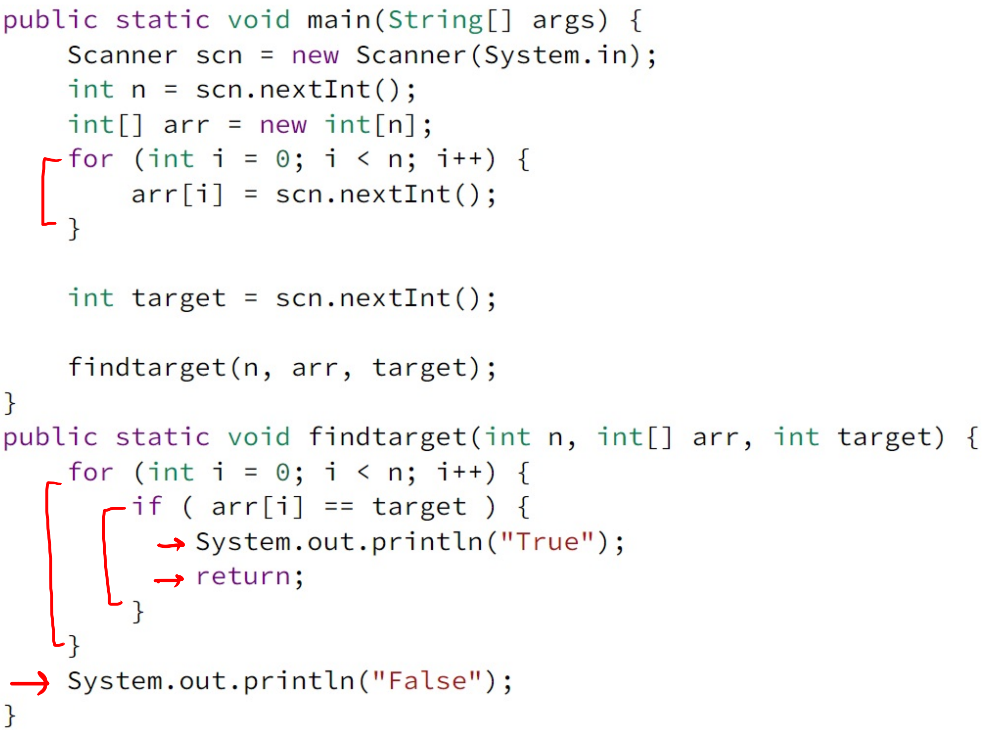
n = 5

arr = 1 2 3 4 5

x = 3

```
for( int i=0; i<n; i++){
    if( arr[i] == x){
        Syso ("True");
    }
}
Syso ("False");
```

Code



```
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();

    findtarget(n, arr, target);
}
public static void findtarget(int n, int[] arr, int target) {
    for (int i = 0; i < n; i++) {
        if ( arr[i] == target ) {
            System.out.println("True");
            return;
        }
    }
    System.out.println("False");
}
```

$arr = [\underset{0}{5}, \underset{1}{4}, \underset{2}{3}, \underset{3}{-2}, \underset{4}{0}, \underset{5}{7}]$

$target = -1$

$i=0$, $(5 == -1)$ X

$i=1$, $(4 == -1)$ X

$i=2$, $(3 == -1)$ X

$i=3$, $(-2 == -1)$ X

$i=4$, $(0 == -1)$ X

$i=5$, $(7 == -1)$ X

$i=6$

False

Print first index of x in array

n = 7

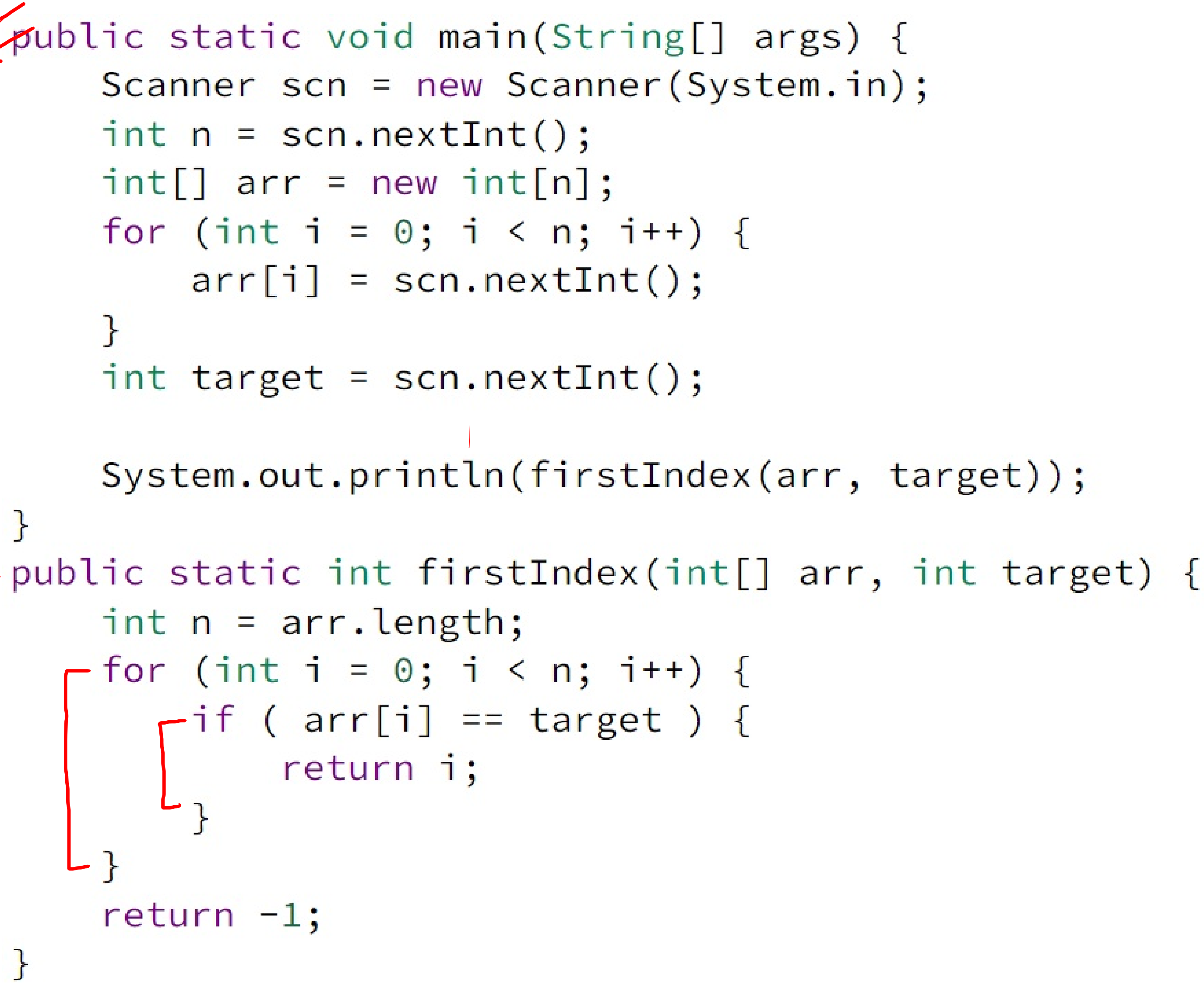
arr = [5 , -2 , 3 , -2 , 3 , 5 , 3]

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

target = 3

$\boxed{\text{ans} = 2}$

Code



```
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();

    System.out.println(firstIndex(arr, target));
}
public static int firstIndex(int[] arr, int target) {
    int n = arr.length;
    for (int i = 0; i < n; i++) {
        if ( arr[i] == target ) {
            return i;
        }
    }
    return -1;
}
```

Print First NON MATCHING NUMBER

n= 6

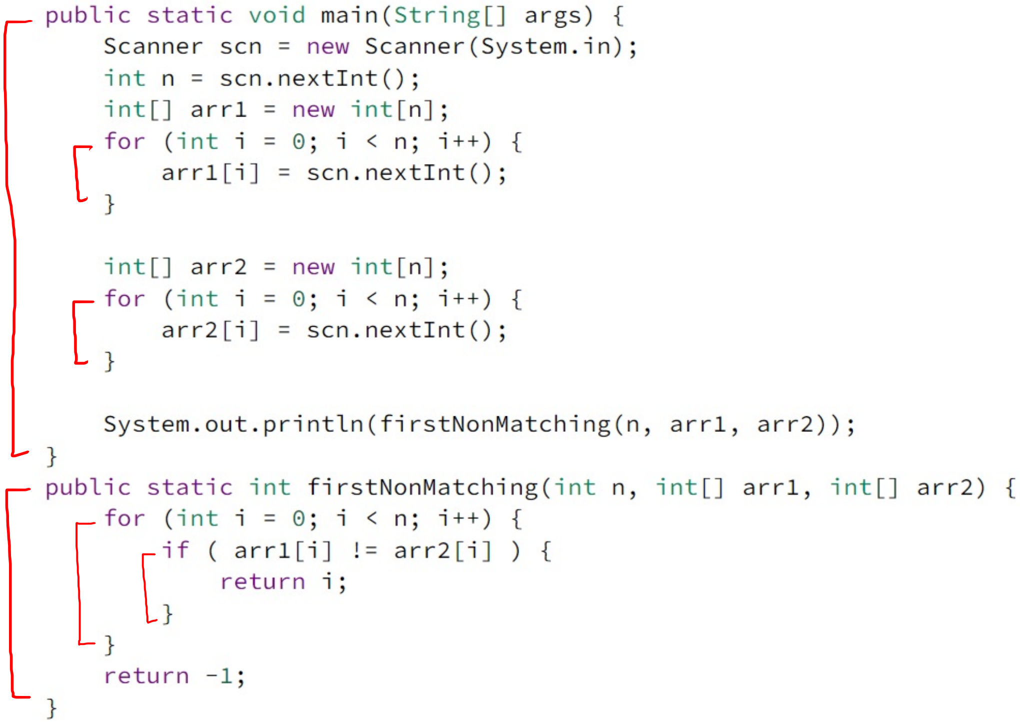
Arr1 =

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

Arr2 =

3	-7	4	8	3	4
0	1	2	3	4	5

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();
    }

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

Note:-

if a function is return type:-

then it must provide some answer back (no matter what)

Note:- if you have no answer then we can return -1 as an answer