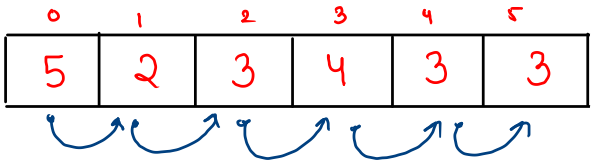
Print first index of x in array

n = 6

arr =



key = 3

ans = 2

Note:-

arr[i]

current element

---

psudo code

1) traverse in array from start to end

1.1) check if curr. element is equal to key

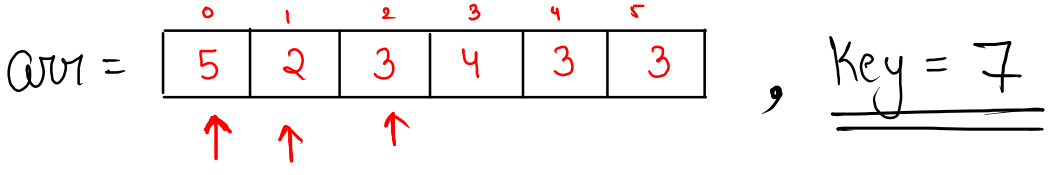
1.1.1) return index

2) return -1

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

main logic



i=0, (5 == 3) ✗

i=1, (2 == 3) ✗

i=2, (3 == 3) ✓

Print First NON MATCHING NUMBER

$n = 5$

arr1 =

5	3	2	7	-2
0	1	2	3	4

arr2 =

5	3	2	-2	7
0	1	2	3	4

ans = 3

psudo code

1) traverse from start to end

1.1) check if curr ele. of arr1 is not equal to curr ele. of arr2

1.1.1) return curr. index

2) return -1

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

camel casing :- kunalSuri ←

snake casing :- kunal_suri ←

→ Updating array elements

Sum of all Elements of Array

n = 5

arr =

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

ans = 3

psudo code

1) create variable sum with value zero
2) traverse in array from start to end
   2.1) update sum by adding curr element
   (sum += arr[i])
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();
    }
    int ans = sumOfArray(arr, n);
    System.out.println(ans);
}
public static int sumOfArray(int[] arr, int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum = sum + arr[i];
    }
    return sum;
}
```

GKSTR35 Count_Even

n = 5

arr =

5	2	4	3	2
0	1	2	3	4

ans = 3

---

psudo code

1) declare count with value zero
2) traverse in array from start to end
   2.1) check if curr. element is even
      2.1.1) increment count by 1
3) return count

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();
    }
    int ans = countEven(arr);
    System.out.println(ans);
}
public static int countEven(int[] arr) {
    int count = 0;
    for (int i = 0; i < arr.length; i++) {
        if ( arr[i] % 2 == 0 ) {
            count++;
        }
    }
    return count;
}
```