```
import java.io.*;
import java.util.*;

public class Solution {

    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();
        for(int i = 0; i < n; i ++) {
            if(arr[i] == key) {
                System.out.println(i);
                return;
            }
        }
        System.out.println(-1);
    }
}
```

124 ⇒ 44

→ [5, 2, 1, 2, 5]

Key = 3

①

1022

S = 3

① → first occ of key in the array

② if there is no occ →

Finding in an array → Linear Search

[5, 5, 2, 7, 1, 2, 1, 7, 3, 6, 1, 3]

x = 1

24 → 2 → 23

3

37 : → 40

2 n−1

2 23

2 to 12

13 23

24

n

24

13

26

13 × 2

21

12

12 × 2 = 24

14

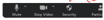
28

14 × 3

14 × 4

2 to n/2

n/2 + 1 → n−1



2 to n−1

2 to n/2

2 to √n

i ≤ √n

i² <= n

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        for(int i = 2; i <= n; i ++) {
            if(isPrime(i) ~~== true~~) {
                System.out.print(i + " ");
            }
        }
    }

    public static boolean isPrime(int n) {
        for(int i = 2; i*i <= n; i ++) {
            if(n % i == 0) return false;
        }
        return true;
    }
}
```

119

① n → 1 to n → print all primes

2 √n n

i² <= n

i <= √n

i² <= n

i ≤ √n

14

8 × 1 = 8

8 × 2 = 16

2

3

4

5

6

7

18

9

18

9 × 2

9 × 2

22

36

① 2 to n−1

② 2 to n/2

③ 2 to √n

36

1 × 18 18 × 2

3 × 12 12 × 3

4 × 9 9 × 4

6 × 6

√36 = 6

≤ √n

√24

4.8

≤ 4

n 24

1 × 24 24 × 1

2 × 12 12 × 2

3 × 8 8 × 3

4 × 6 6 × 4

6

8

12

24

5

2−3

9 : 08

i <= √n

i*i <= n

i² <= n

i <= √n

i

n/i

n = 24

i = 2 → 2

n/i = 24/2

12

```
int l = Integer.MIN_VALUE, sl = Integer.MIN_VALUE;
for(int i = 0; i < n; i ++) {
    int ele = arr[i];
    if(ele > l) {
        sl = l;
        l = ele;
    } else if(ele > sl && ele != l) {
        sl = ele;
    }
}
System.out.println(sl);
```

[4, 6, 9, 1, 2, 8, 9, 9]

ele = 4

false i = 0

8

l = ~~4~~ ~~6~~ 9

sl = ~~−∞~~ ~~−∞~~ ~~4~~ ~~6~~ 8

l = ~~−∞~~ ~~4~~ ~~6~~ 9 9

sl = ~~−∞~~ ~~−∞~~ ~~4~~ ~~4~~ 8

8

8 ≥ 4

[1, 7, 2, 4, 9, 6, 3, 8, 0, 5]

l = ⇒ l = ~~7~~ 9

sl = ⇒ sl = ~~7~~ 8

→ 8

int max = Integer.MIN_VALUE;

−2³¹ → BITS

① [−1, −3, −5, −2, −6, −10]

Integer.MIN_VALUE

−∞

max = −1

arr[0]