Arrays → Basics

memory managent

Qn/

double[] arr = new double[n];

↳ size

else ↳ ↳ ↳ ↳ ↳

arr[1] == arr[2]

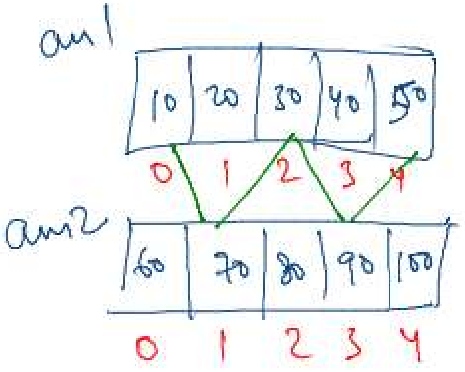
⇓

false

↳ print return

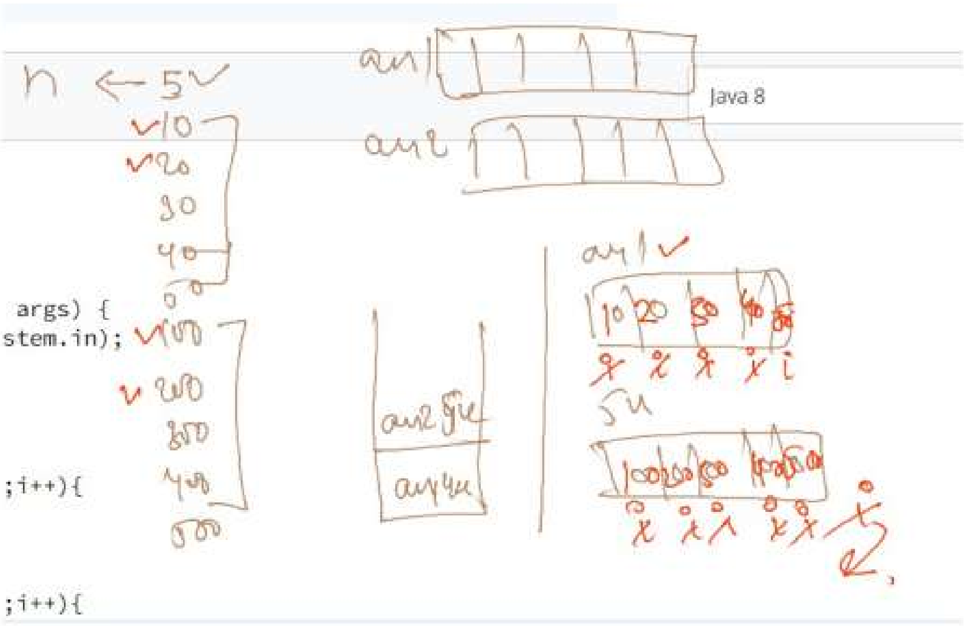
Qn n-value to declare length of arrays

n | arr1   n | arr2

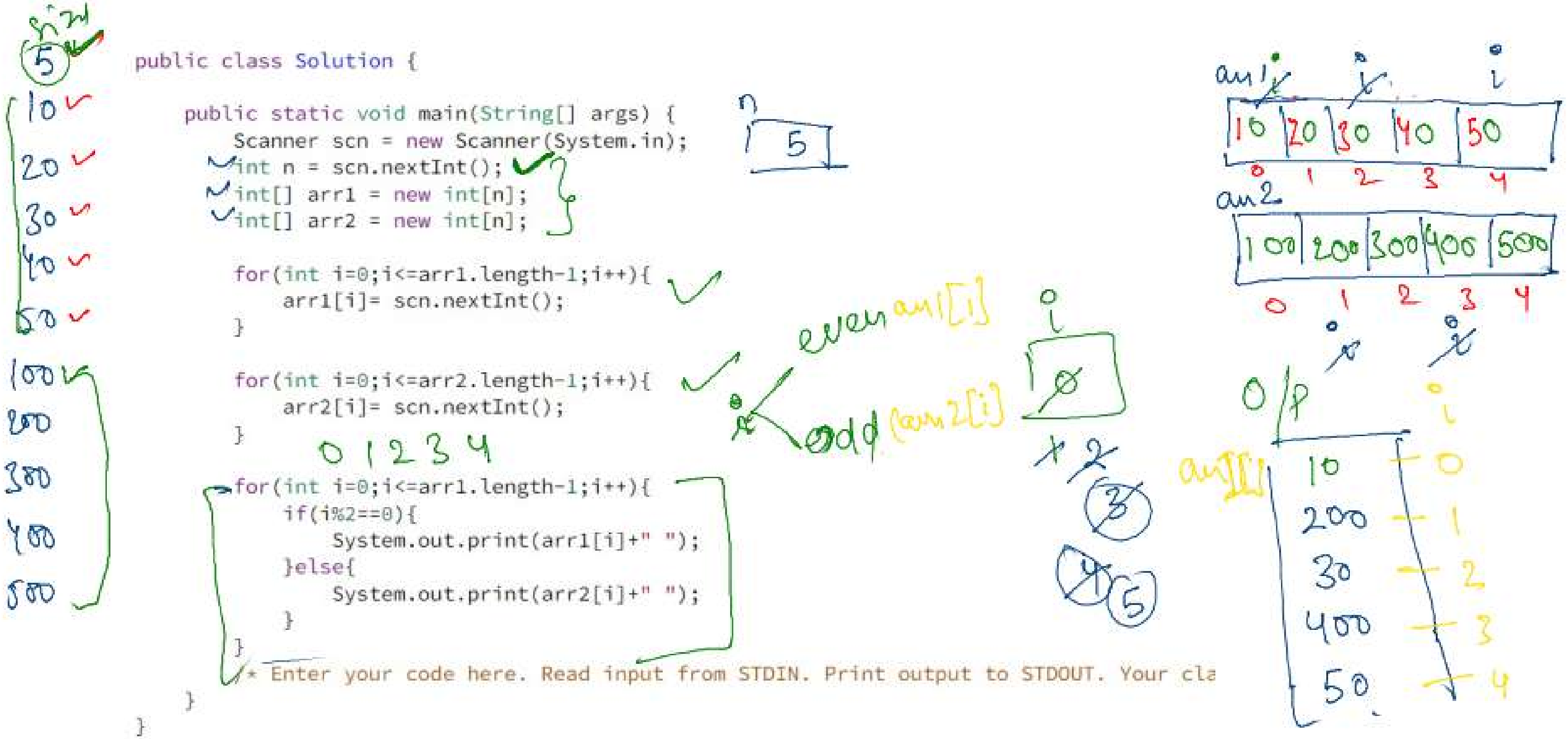


o/p   Index

arr1 ← 10 — 0
arr2 ← 70 — 1
arr1 ← 30 — 2
arr2 ← 90 — 3
arr1 ← 50 — 4



```
    args) {
stem.in);

;i++){

;i++){
```



```
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[] arr1 = new int[n];
        int[] arr2 = new int[n];

        for(int i=0;i<=arr1.length-1;i++){
            arr1[i]= scn.nextInt();
        }

        for(int i=0;i<=arr2.length-1;i++){
            arr2[i]= scn.nextInt();
        }

        for(int i=0;i<=arr1.length-1;i++){
            if(i%2==0){
                System.out.print(arr1[i]+" ");
            }else{
                System.out.print(arr2[i]+" ");
            }
        }
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your cla
    }
}
```

Seaching

Qury find

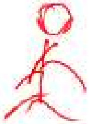
search space

arr =

10	20	30	40	~~21~~	9	40	50	51
0	1	2	3	4	5	6	7	8

target

21



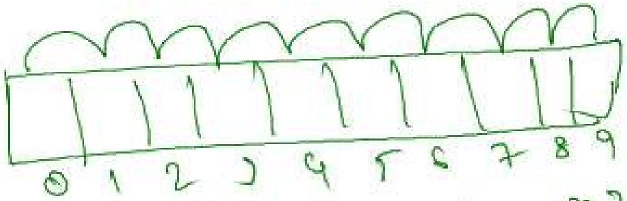
→ younger $000/sis → police

(name, photo, id) (target)

last location (search space)

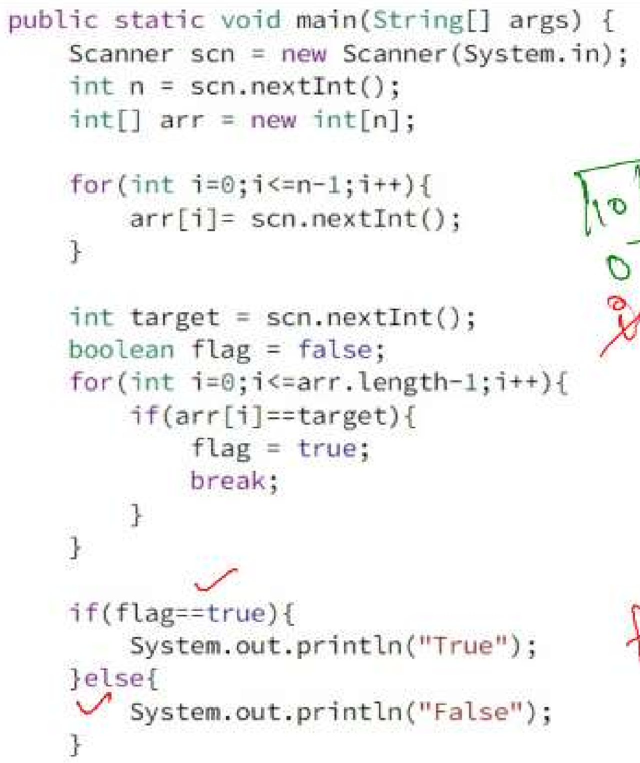
True/ false

arr



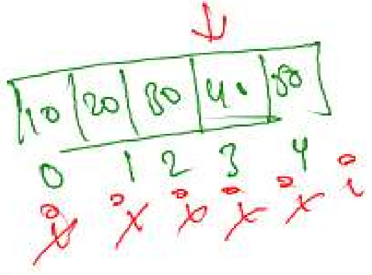
⇒ linear Search

```
for (i=0 to lngth-1){
    if (arr[i] = = target){
        syso(True)
    }else syso(false);
}
```



```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-1;i++){
        arr[i]= scn.nextInt();
    }

    int target = scn.nextInt();
    boolean flag = false;
    for(int i=0;i<=arr.length-1;i++){
        if(arr[i]==target){
            flag = true;
            break;
        }
    }

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

Target = 40



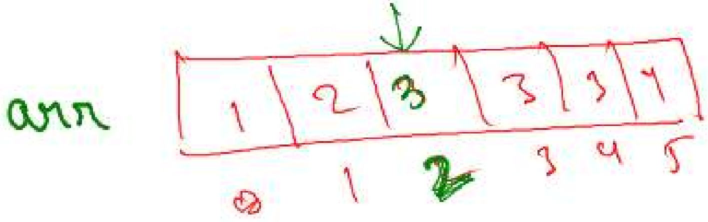
True

false True

~~false~~

Que first index.

target = 3

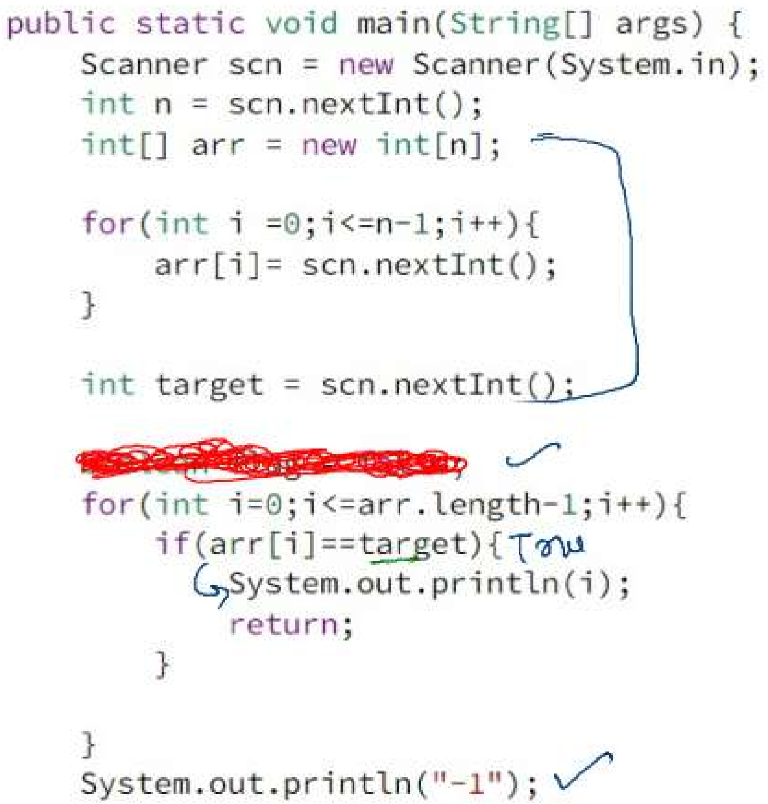


o/p = 2

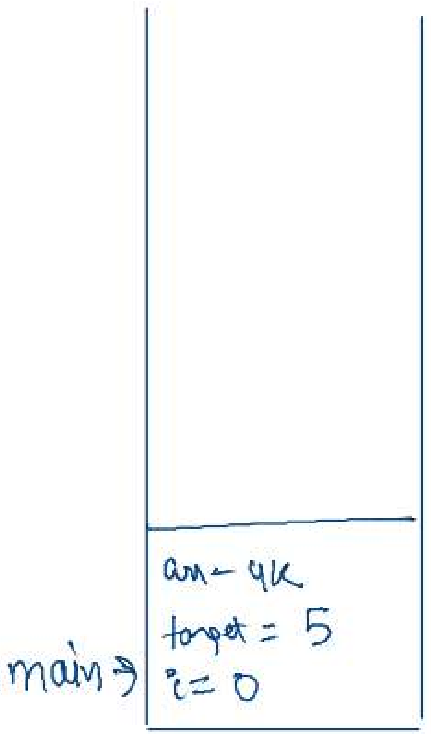
break | return

↳ will pop out your function from stack

↳ break will stop executing that block



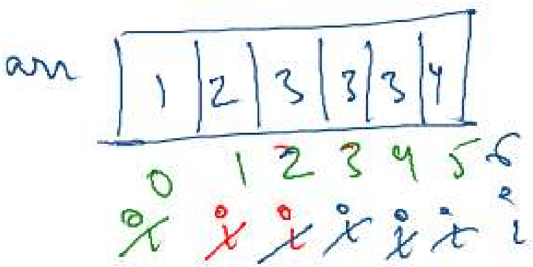
```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];

    for(int i =0;i<=n-1;i++){
        arr[i]= scn.nextInt();
    }

    int target = scn.nextInt();

    [redacted] ✓
    for(int i=0;i<=arr.length-1;i++){
        if(arr[i]==target){ True
          System.out.println(i);
            return;
        }

    }
    System.out.println("-1"); ✓
```



o/p 2

$\boxed{3==3}$

-1



For last index

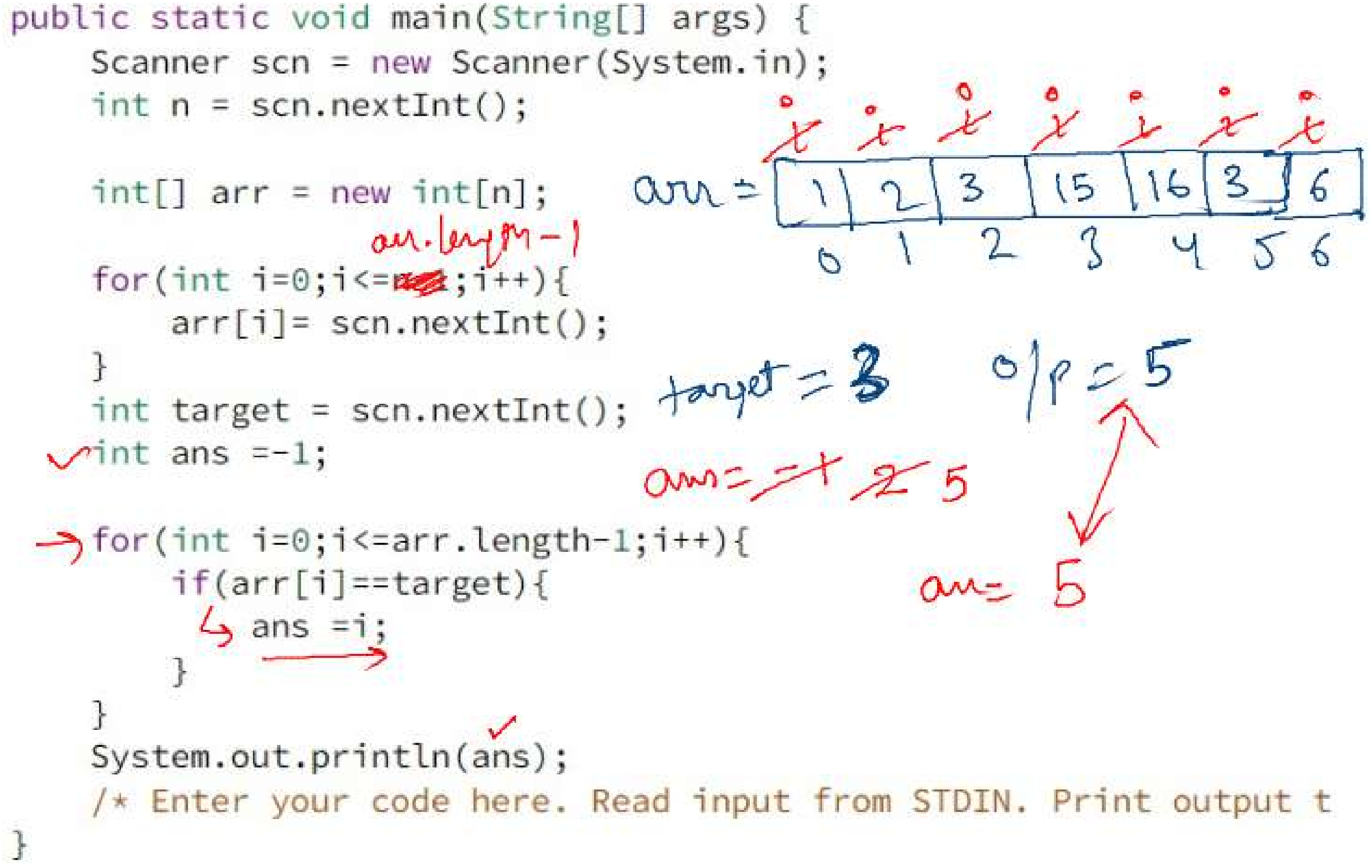
targ = 3

arr | 1 | 2 | 3 | 4 | 4 | 3 | 3 | 5 |

0 1 2 3 4 5 6 7

o/p = 3

sol



```
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<=[illegible];i++){
        arr[i]= scn.nextInt();
    }
    int target = scn.nextInt();
    int ans =-1;

    for(int i=0;i<=arr.length-1;i++){
        if(arr[i]==target){
            ans =i;
        }
    }
    System.out.println(ans);
    /* Enter your code here. Read input from STDIN. Print output t
}
```

arr.length-1

i i i i i i i i

arr = | 1 | 2 | 3 | 15 | 16 | 3 | 6 |

0 1 2 3 4 5 6

target = 3

o/p = 5

ans = ~~-1~~ ~~2~~ 5

ans = 5