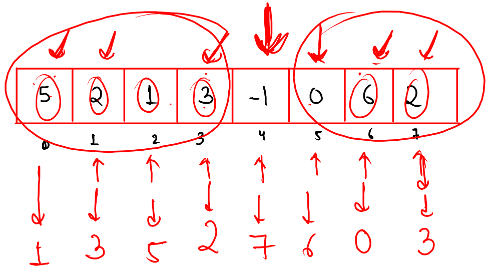
```java
// main logic
public static void greaterThanMe(int[] arr, int n) {    // declaration of function

    for (int i = 0; i < n; i++) {
        int count = 0;
        for (int j = 0; j < n; j++) {
            if ( i != j && arr[j] > arr[i] ) {
                count++;
            }
        }
        System.out.print(count + " ");
    }
}
```

i=0, 5 count=~~0~~1
i=1, 2, count=0

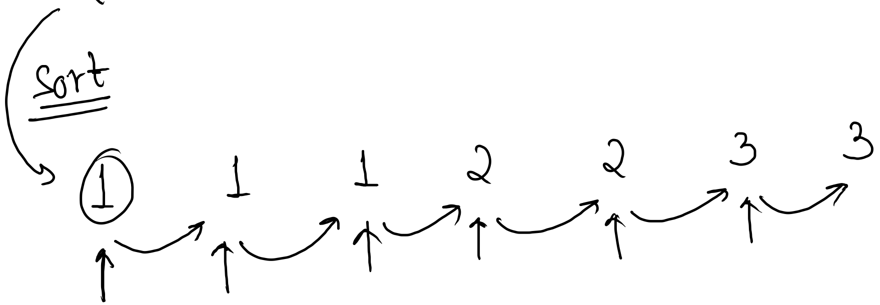


```java
public class Solution {

    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();
        }

        greaterThanMe(arr, n);  // calling of function
    }

    // main logic
    public static void greaterThanMe(int[] arr, int n) {    // declaration of function

        for (int i = 0; i < n; i++) {
            int count = 0;
            for (int j = 0; j < n; j++) {
                if ( i != j && arr[j] > arr[i] ) {
                    count++;
                }
            }
            System.out.print(count + " ");
        }
    }
}
```

Max Count 3

ex:-

2 1 1 3 2 1 3

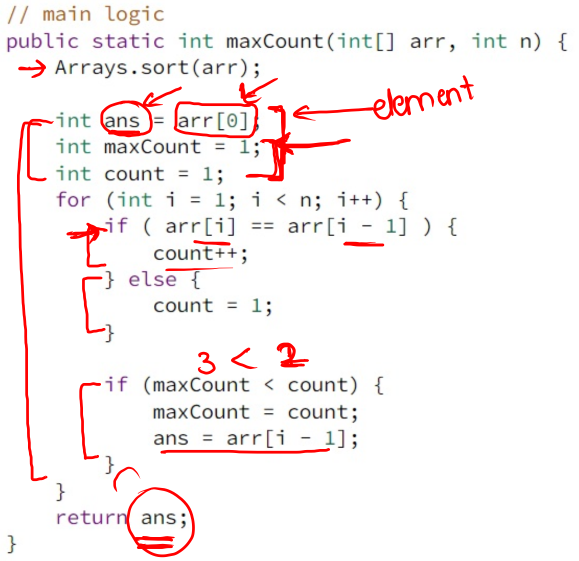


count = ~~1~~ ~~2~~ 3
= ~~1~~ 2
= ~~1~~ 2

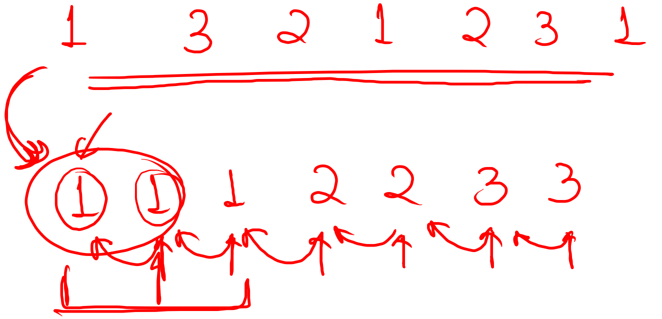
max

approch
↳ sort
↳ loop

```java
// main logic
public static int maxCount(int[] arr, int n) {
    Arrays.sort(arr);

    int ans = arr[0];
    int maxCount = 1;
    int count = 1;
    for (int i = 1; i < n; i++) {
        if ( arr[i] == arr[i - 1] ) {
            count++;
        } else {
            count = 1;
        }

        if (maxCount < count) {
            maxCount = count;
            ans = arr[i - 1];
        }
    }
    return ans;
}
```



```
// main logic
public static int maxCount(int[] arr, int n) {
    Arrays.sort(arr);

    int ans = arr[0];
    int maxCount = 1;
    int count = 1;
    for (int i = 1; i < n; i++) {
        if ( arr[i] == arr[i - 1] ) {
            count++;
        } else {
            count = 1;
        }

        if (maxCount < count) {
            maxCount = count;
            ans = arr[i - 1];
        }
    }
    return ans;
}
```



count = ~~1~~ ~~2~~ ~~3~~ ~~1~~ ~~2~~ ~~1~~ 2

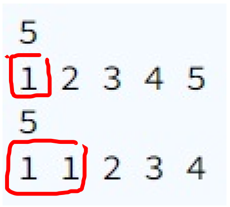
max Count = ~~1~~ ~~2~~ 3

ans = ~~1~~ ~~1~~ 1

642 642 642

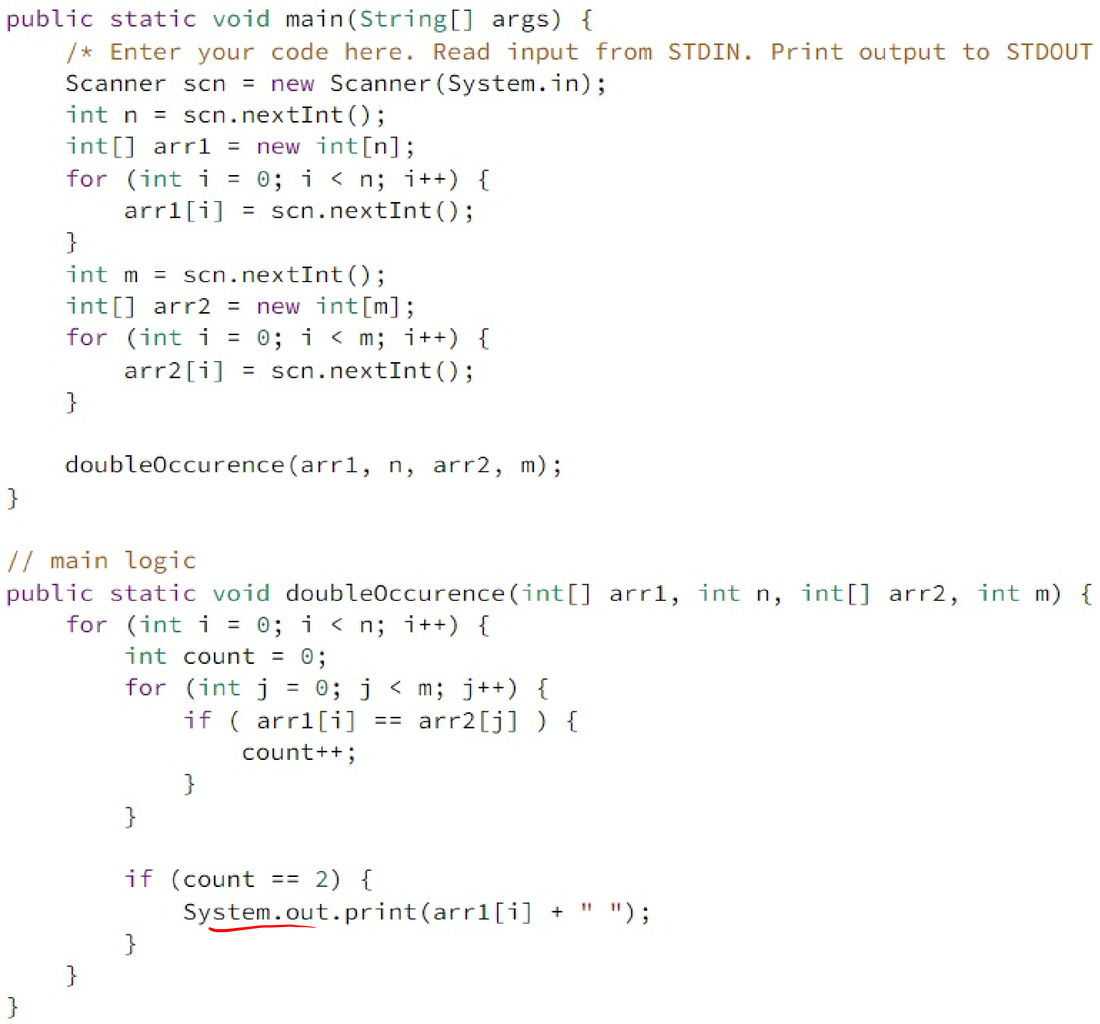
ans = 642

Double Occurence



```
5
1 2 3 4 5
5
1 1 2 3 4
```

ques :- print all elements of 1st array which occur twice in 2nd array



```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT
    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 m = scn.nextInt();
    int[] arr2 = new int[m];
    for (int i = 0; i < m; i++) {
        arr2[i] = scn.nextInt();
    }

    doubleOccurence(arr1, n, arr2, m);
}

// main logic
public static void doubleOccurence(int[] arr1, int n, int[] arr2, int m) {
    for (int i = 0; i < n; i++) {
        int count = 0;
        for (int j = 0; j < m; j++) {
            if ( arr1[i] == arr2[j] ) {
                count++;
            }
        }

        if (count == 2) {
            System.out.print(arr1[i] + " ");
        }
    }
}
```