

# Max Count 3

Problem

Submissions

Leaderboard

Discussions

Take an array of size **n** with integer elements. And **Print** an element in the array which occurs for the **maximum** number of times.

eg.

count: 2  
val = 3

1	2	3	1	1	2	3	2	2
0	1	2	3	4	5	6	7	8



count > final count

3 > 0  
4 > 3  
2 > 4

ans = 2

## Sample Input 0

```
7
1 1 1 2 2 3 3
```

## Sample Output 0

```
1
```

final Ans = 1  
final Count = 4

final count = ~~0~~ ~~3~~ 4  
 \* final val = ~~0~~ ~~1~~ 2

1	2	3	1	1	2	3	2	2
0	1	2	3	4	5	6	7	8

→

(freq) count = 3  
 val = 1

1 ... (3) ✓

if (count > final count)  
 3 > 0

4 > 3  
 (2 > 1) ✗

↑  
 i = 2

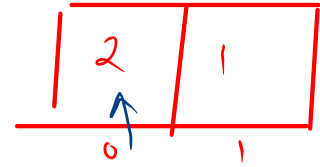
(3 > 1)

```

4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13
14        //logic
15        int ansVal = 0;
16        int ansFreq = 0;
17        for(int i = 0; i < n; i++){
18            int freq = 0; //current freq or count
19            for(int j = 0; j < n; j++){
20                if(A[i] == A[j]){
21                    freq++;
22                }
23            }
24            if(freq > ansFreq){
25                ansVal = A[i];
26                ansFreq = freq;
27            }
28        }
29        System.out.println(ansVal);
30    }
31 }

```

eg.



n=2

ansVal = 2

ansFreq = 1

i = 0      0 < 2  
1          1 < 2 ✓

freq = 1

1 > 1

# Find Duplicate 3

$n = 5$

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Take an array of size  $n$  with integer input. And Print **"true"** if the array contains a duplicate element and print **"false"**, if the array doesn't contain a duplicate element.

1	2	3	4	1
0	1	2	3	4

↑  
i

ans = true

(j)

$A[i] == A[j]$

$i == j$

$$\left[ \begin{array}{l} i \neq j \ \&\& \ A[i] == A[j] \\ \hline \text{true} \end{array} \right]$$

```

13
14 //logic
15 boolean ans = false;
16 /*assume
17 ans false means there is no duplicates
18 Now, I will try to find duplicate
19 */
20 for(int i = 0; i < n; i++){
21     if(ans == true){
22         break;
23     }
24     for(int j = 0; j < n; j++){
25         if(i != j && A[i] == A[j]){
26             ans = true;
27             break;
28         }
29     }
30 }
31 System.out.println(ans);
32
33 }
34 }

```

ans = ~~F~~ T

1	2	2	3	...
0	1	2	3	

→ (i)

j

1 == 2

1 != 2 ✓

```
5 public static boolean checkDuplicate(int [] A){
6     int n = A.length;
7     for(int i = 0; i < n; i++){
8         for(int j = 0; j < n; j++){
9             if(i != j && A[i] == A[j]){
10                 return true;
11             }
12         }
13     }
14     return false;
15 }
```

# Double Occurence

Problem

Submissions

Leaderboard

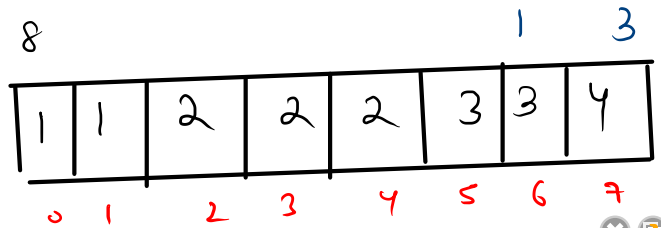
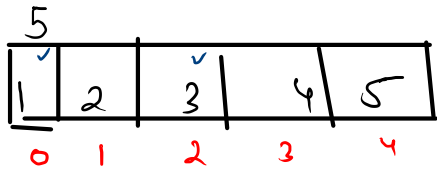
Discussions

Given an array of size  $n$  with unique integer elements. And then take  $m$  as an integer input. Declare the second array of size  $m$  that stores values of int data-type. Then take  $m$  integer inputs and store them in the array one by one.

Then print all the elements of the first array which occur exactly twice in the second array.

o/p

eg.



1

count = 3

A[i] == B[j]  
c++

count == 2

```

1 import java.io.*;
2 import java.util.*;
3 public class Solution {
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int n = scn.nextInt();
7         int [] A = new int[n];
8         for(int i = 0 ; i < n; i++){
9             A[i] = scn.nextInt();
10        }
11        int m = scn.nextInt();
12        int [] B = new int[m];
13        for(int i = 0 ; i < m; i++){
14            B[i] = scn.nextInt();
15        }
16        //logic
17        for(int i = 0; i < n; i++){
18            int count = 0;
19            for(int j = 0; j < m; j++){
20                if(A[i] == B[j]){
21                    count++;
22                }
23            }
24            if(count == 2){
25                System.out.print(A[i] + " ");
26            }
27        }
28    }
29 }

```

A → 2 1  
0 1

i

n=2

m=3

B → 2 2 1  
0 1 2

j

count = 1

2



maximum difference between the two elements

Given an array `arr[]` of integers, find out the maximum difference between any two elements such that larger element appears after the smaller number.

abs-

Sample Input 0

7  
2 3 10 6 4 8 1

Sample Output 0

8

Diagram illustrating the calculation of absolute differences between elements in the array [2, 3, 10, 6, 4, 8, 1]. The elements are labeled s (smaller) and l (larger). The differences are calculated as follows:

s	l	abs(l - s)
2	3	1
2	10	8
2	6	4
2	4	2
2	8	6
2	1	-1
3	10	7
3	6	3
3	4	1
3	8	5
3	1	
10	6	-4
10	4	-6
10	8	-2
10	1	-9
6	4	-2
6	8	2
6	1	-5
4	8	4
4	1	-3
8	1	-7

max  $l - s$   
 $A[j] - A[i]$

```
13  
14 //logic  
15 int ans = 0;  
16 for(int i = 0; i < n; i++){  
17     for(int j = i + 1; j < n; j++){  
18         ans = Math.max(A[j]-A[i], ans);  
19     }  
20 }  
21 System.out.println(ans);  
22 }  
23 }
```