

## Candidate Report: Anonymous

Test Name:

[Summary](#)[Timeline](#)

### Test Score

55 out of 100 points

# 55%

### Tasks in Test

OddOccurrencesInArray  
Submitted in: Java 8

Time Spent ⓘ

27 min

Task Score

55%

### TASKS DETAILS

EASY

#### 1. [OddOccurrencesInArray](#)

Find value that occurs in odd number of elements.

Task Score

55%

Correctness

80%

Performance

25%

### Task description

A non-empty array A consisting of N integers is given. The array contains an odd number of elements, and each element of the array can be paired with another element that has the same value, except for one element that is left unpaired.

For example, in array A such that:

```
A[0] = 9  A[1] = 3  A[2] = 9
A[3] = 3  A[4] = 9  A[5] = 7
A[6] = 9
```

- the elements at indexes 0 and 2 have value 9,
- the elements at indexes 1 and 3 have value 3,
- the elements at indexes 4 and 6 have value 9,
- the element at index 5 has value 7 and is unpaired.

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given an array A consisting of N integers fulfilling the above conditions, returns the value of the unpaired element.

For example, given array A such that:

```
A[0] = 9  A[1] = 3  A[2] = 9
A[3] = 3  A[4] = 9  A[5] = 7
A[6] = 9
```

the function should return 7, as explained in the example above.

Write an **efficient** algorithm for the following assumptions:

- N is an odd integer within the range [1..1,000,000];

### Solution

Programming language used: Java 8

Total time used: 27 minutes ⓘ

Effective time used: 27 minutes ⓘ

Notes: *not defined yet*

### Task timeline ⓘ



17:34:30

18:01:30

Code: 18:01:30 UTC, java, final,  
score: 55[show code in pop-up](#)

```
1 // you can also use imports, for example:
2 // import java.util.*;
3
4 // you can write to stdout for debugging purposes, e.g.
5 // System.out.println("this is a debug message");
6
7 class Solution {
8     public static int max = 1000000;
9     public static boolean[] mem0 = new boolean[max];
```

- each element of array A is an integer within the range [1..1,000,000,000];
- all but one of the values in A occur an even number of times.

Copyright 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
10      public static boolean[] mem1 = new boolean[max];
11
12      public static int solution(int[] a) {
13          if (a.length == 1)
14              return a[0];
15
16          for (int i = 0; i < a.length; i++)
17              if (mem0[a[i]]) {
18                  mem1[a[i]] = true;
19              } else {
20                  mem0[a[i]] = true;
21              };
22
23          for (int i = 0; i < max; i++)
24              if (mem0[i] && !mem1[i])
25                  return i;
26
27          return -1;
28      }
29  }
```

Analysis summary

The following issues have been detected: wrong answers, runtime errors.

Analysis ?

Example tests	
▶ example1 example test	✓ OK
Correctness tests	
▶ simple1 simple test n=5	✓ OK
▶ simple2 simple test n=11	✓ OK
▶ extreme_single_item [42]	✓ OK
▶ small1 small random test n=201	✗ WRONG ANSWER got -1 expected 42
▶ small2 small random test n=601	✓ OK
Performance tests	
▶ medium1 medium random test n=2,001	✓ OK
▶ medium2 medium random test n=100,003	✗ RUNTIME ERROR tested program terminated with exit code 1
▶ big1 big random test n=999,999, multiple repetitions	✗ WRONG ANSWER got -1 expected 700
▶ big2 big random test n=999,999	✗ RUNTIME ERROR tested program terminated with exit code 1

PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.