

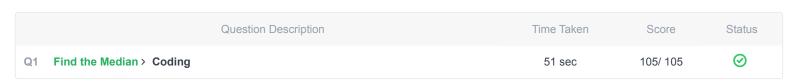
## Mock Test > tunahan.erken@outlook.com

**Full Name:** Tunahan ERKEN Email: tunahan.erken@outlook.com Test Name: **Mock Test** Taken On: 4 Jun 2024 12:10:12 IST Time Taken: 1 min 3 sec/ 10 min Invited by: Ankush 4 Jun 2024 12:10:02 IST Invited on: Skills Score: Tags Score: Algorithms 105/105 Core CS 105/105 Easy 105/105 Problem Solving 105/105 Search 105/105 105/105 Sorting

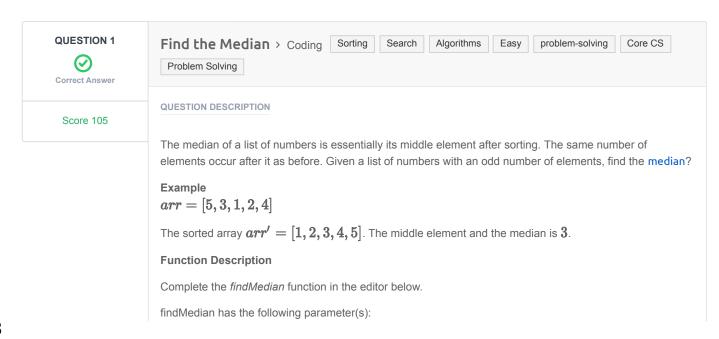
100% scored in Mock Test in 1 min 3 sec on 4 Jun 2024 12:10:12 IST 105/105

# **Recruiter/Team Comments:**

No Comments.



problem-solving 105/105



• int arr[n]: an unsorted array of integers

#### Returns

int: the median of the array

## **Input Format**

The first line contains the integer n, the size of arr.

The second line contains  $m{n}$  space-separated integers  $m{arr}[m{i}]$ 

### Constraints

- $1 \le n \le 1000001$
- **n** is odd
- $-10000 \le arr[i] \le 10000$

### Sample Input 0

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

### Sample Output 0

3

### **Explanation 0**

The sorted arr = [0, 1, 2, 3, 4, 5, 6]. It's middle element is at arr[3] = 3.

#### **CANDIDATE ANSWER**

# Language used: C#

```
1 using System.CodeDom.Compiler;
 2 using System.Collections.Generic;
 3 using System.Collections;
4 using System.ComponentModel;
 5 using System. Diagnostics. CodeAnalysis;
6 using System. Globalization;
7 using System.IO;
8 using System.Linq;
9 using System.Reflection;
10 using System.Runtime.Serialization;
11 using System.Text.RegularExpressions;
12 using System. Text;
13 using System;
14
17 class Result
18 {
      * Complete the 'findMedian' function below.
      * The function is expected to return an INTEGER.
      * The function accepts INTEGER_ARRAY arr as parameter.
       public static int findMedian(List<int> arr)
              arr.Sort();
```

```
int returnValue = (arr.Count - 1) / 2;
              return arr[returnValue];
36 }
38 class Solution
39 {
     public static void Main(string[] args)
           TextWriter textWriter = new
43 StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT_PATH"),
              int n = Convert.ToInt32(Console.ReadLine().Trim());
47
              while (n < 1 \mid | n > 1000001 \mid | n % 2 == 0)
49
                  n = Convert.ToInt32(Console.ReadLine().Trim());
               List<int> arr = Console.ReadLine().TrimEnd().Split('
54 ').ToList().Select(arrTemp => Convert.ToInt32(arrTemp)).ToList();
              while (arr.Count != n)
                  arr = Console.ReadLine().TrimEnd().Split('
59 ').ToList().Select(arrTemp => Convert.ToInt32(arrTemp)).ToList();
               int result = Result.findMedian(arr);
          textWriter.WriteLine(result);
          textWriter.Flush();
           textWriter.Close();
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0481 sec	22.9 KB
Testcase 2	Easy	Hidden case	Success	35	0.0708 sec	24.4 KB
Testcase 3	Easy	Hidden case	Success	35	0.064 sec	24.3 KB
Testcase 4	Easy	Hidden case	Success	35	0.0669 sec	32.8 KB

No Comments

PDF generated at: 4 Jun 2024 06:42:51 UTC