

Full Name: ToYoNiX

Email: toyonix.assemmohamed.2005@gmail.com

Test Name: Mock Test

**Taken On:** 6 Sep 2023 16:39:40 IST

Time Taken: 5 min 54 sec/ 10 min

Invited by: Ankush

Invited on: 6 Sep 2023 16:29:07 IST

Skills Score:

Tags Score: Algorithms 105/105

Core CS 105/105

Easy 105/105

Problem Solving 105/105

Search 105/105

Sorting 105/105

problem-solving 105/105

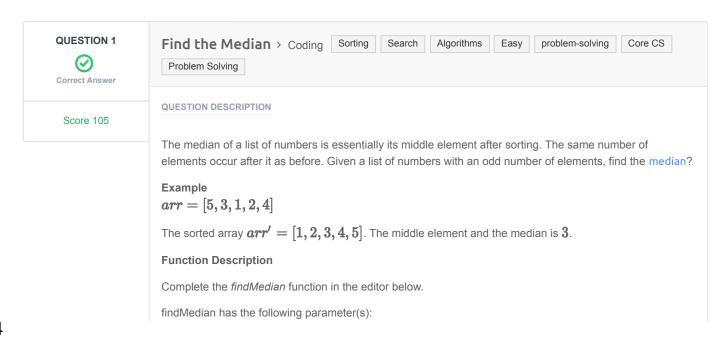


scored in **Mock Test** in 5 min 54 sec on 6 Sep 2023 16:39:40 IST

# **Recruiter/Team Comments:**

No Comments.





• *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++14

```
1 #include <bits/stdc++.h>
 2 #include <algorithm>
4 using namespace std;
6 string ltrim(const string &);
7 string rtrim(const string &);
8 vector<string> split(const string &);
12 /*
* Complete the 'findMedian' function below.
14
* The function is expected to return an INTEGER.
* The function accepts INTEGER ARRAY arr as parameter.
19 int findMedian(int *arr, int n) {
     sort(arr, arr + n);
     return arr[n / 2];
22 }
24 int main()
25 {
     ofstream fout(getenv("OUTPUT PATH"));
     string n temp;
     getline(cin, n_temp);
```

```
int n = stoi(ltrim(rtrim(n_temp)));
       string arr_temp_temp;
       getline(cin, arr temp temp);
       vector<string> arr temp = split(rtrim(arr temp temp));
       int arr[n];
       for (int i = 0; i < n; i++) {
           int arr_item = stoi(arr_temp[i]);
           arr[i] = arr item;
       }
       int result = findMedian(arr, n);
47
      fout << result << "\n";
      fout.close();
      return 0;
53 }
55 string ltrim(const string &str) {
      string s(str);
      s.erase(
           s.begin(),
           find if(s.begin(), s.end(), not1(ptr fun<int, int>(isspace)))
      return s;
64 }
66 string rtrim(const string &str) {
      string s(str);
       s.erase(
           find if(s.rbegin(), s.rend(), not1(ptr fun<int, int>
71 (isspace))).base(),
           s.end()
      );
      return s;
76 }
78 vector<string> split(const string &str) {
      vector<string> tokens;
      string::size_type start = 0;
       string::size_type end = 0;
       while ((end = str.find(" ", start)) != string::npos) {
           tokens.push back(str.substr(start, end - start));
           start = end + 1;
       }
       tokens.push back(str.substr(start));
       return tokens;
93 }
```

	TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
	Testcase 1	Easy	Sample case	Success	0	0.0267 sec	8.88 KB
	Testcase 2	Easy	Hidden case	Success	35	0.0359 sec	9.07 KB
	Testcase 3	Easy	Hidden case	Success	35	0.0339 sec	9.14 KB
	Testcase 4	Easy	Hidden case	Success	35	0.0444 sec	13.1 KB
No Comments							

PDF generated at: 6 Sep 2023 11:17:07 UTC