Full Name:	Sean Sanii
Email:	seansanii@outlook.com
Test Name:	**Mock Test**
Taken On:	30 Jun 2025 16:40:23 IST
Time Taken:	3 min 48 sec/ 15 min
Invited by:	Ankush
Invited on:	30 Jun 2025 16:40:17 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

100%

105/105

scored in **Mock Test** in 3 min 48 sec on 30 Jun 2025 16:40:23 IST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Find the Median > **Coding**	3 min 41 sec	105/ 105	✓

QUESTION 1

Correct Answer

Score 105

Find the Median > Coding

Sorting | Search | Algorithms | Easy | problem-solving | Core CS | Problem Solving

QUESTION DESCRIPTION

The median of a list of numbers is essentially its middle element after sorting. The same number of elements occur after it as before. Given a list of numbers with an odd number of elements, find the median?

Example

$arr = [5, 3, 1, 2, 4]$

The sorted array $arr' = [1, 2, 3, 4, 5]$. The middle element and the median is 3.

Function Description

Complete the *findMedian* function in the editor below.

findMedian has the following parameter(s):

- *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 n space-separated integers $arr[i]$

Constraints

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

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
2  /*
3   * Complete the 'findMedian' function below.
4   *
5   * The function is expected to return an INTEGER.
6   * The function accepts INTEGER_ARRAY arr as parameter.
7   */
8
9  int findMedian(vector<int> arr) {
10     sort(arr.begin(), arr.end());
11     int middle = arr.size();
12
13     return arr[middle/2];
14 }
15
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0076 sec	8.63 KB
Testcase 2	Easy	Hidden case	Success	35	0.0097 sec	9.04 KB
Testcase 3	Easy	Hidden case	Success	35	0.0127 sec	8.75 KB
Testcase 4	Easy	Hidden case	Success	35	0.0312 sec	13.1 KB

No Comments