

Find the Median

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)

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 $arr = [0, 1, 2, 3, 4, 5, 6]$. Its middle element is at $arr[3] = 3$.

The contest has not yet started. It begins in an hour.

Submissions: 0

Max Score: 0

Difficulty: Easy

Rate This Challenge:



More

C++



```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 string ltrim(const string &);
6 string rtrim(const string &);
7 vector<string> split(const string &);
8
9 /*
10  * Complete the 'findMedian' function below.
11  *
12  * The function is expected to return an INTEGER.
13  * The function accepts INTEGER_ARRAY arr as parameter.
14  */
15
16 int findMedian(vector<int> arr) {
17 }
18
19 int main()
20 {
21     ofstream fout(getenv("OUTPUT_PATH"));
22
23     string n_temp;
24     getline(cin, n_temp);
25
26     int n = stoi(ltrim(rtrim(n_temp)));
27
28     string arr_temp_temp;
29     getline(cin, arr_temp_temp);
30
31     vector<string> arr_temp = split(rtrim(arr_temp_temp));
32
33     vector<int> arr(n);
34
35     for (int i = 0; i < n; i++) {
36         int arr_item = stoi(arr_temp[i]);
37
38         arr[i] = arr_item;
39     }
40
41     int result = findMedian(arr);
42
43     fout << result << "\n";
44
45     fout.close();
46
47     return 0;
48 }
49
50 string ltrim(const string &str) {
```

```
52     string s(str);
53
54     s.erase(
55         s.begin(),
56         find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
57     );
58
59     return s;
60 }
61
62▼ string rtrim(const string &str) {
63     string s(str);
64
65     s.erase(
66         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
67         s.end()
68     );
69
70     return s;
71 }
72
73▼ vector<string> split(const string &str) {
74     vector<string> tokens;
75
76     string::size_type start = 0;
77     string::size_type end = 0;
78
79▼ while ((end = str.find(" ", start)) != string::npos) {
80     tokens.push_back(str.substr(start, end - start));
81
82     start = end + 1;
83 }
84
85 tokens.push_back(str.substr(start));
86
87 return tokens;
88 }
89 }
```

Line: 1 Col: 1

 Upload Code as File Test against custom input Run Code Submit Code