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
Invited on:	4 Jun 2024 12:10:02 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 1 min 3 sec on 4 Jun 2024 12:10:12 IST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Find the Median > **Coding**	51 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#**

```
using System.CodeDom.Compiler;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Diagnostics.CodeAnalysis;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Reflection;
using System.Runtime.Serialization;
using System.Text.RegularExpressions;
using System.Text;
using System;



class Result
{

    /*
     * 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];
    }

}

class Solution
{
    public static void Main(string[] args)
    {
        TextWriter textWriter = new
StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT_PATH"),
true);

            int n = Convert.ToInt32(Console.ReadLine().Trim());

            while (n < 1 || n > 1000001 || n % 2 == 0 )
            {
                n = Convert.ToInt32(Console.ReadLine().Trim());
            }

            List<int> arr = Console.ReadLine().TrimEnd().Split('
').ToList().Select(arrTemp => Convert.ToInt32(arrTemp)).ToList();

            while (arr.Count != n)
            {
                arr = Console.ReadLine().TrimEnd().Split('
').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