**Title**

-------------------------------------------------------------------------------------------------------------

**Find the Median**

-------------------------------------------------------------------------------------------------------------

**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 <= n <= 1000001

n is odd

-10000 <= arr[i] <= 10000

**Sample Input**

7

0 1 2 4 6 5 3

**Sample Output**

3

-------------------------------------------------------------------------------------------------------------

**Code**

-------------------------------------------------------------------------------------------------------------

package main

import(

    "fmt"

    "sort"

)

func main(){

    slc := []int32{0, 1, 2, 4, 6, 5, 3}

    var med int32

    med = findMedian(slc)

    fmt.Println(med)

}

func findMedian(arr[] int32)int32{

    slc:=make([]int,len(arr))

    for \_,v:= range arr{

        slc= append(slc,int(v))

    }

    sort.Sort(sort.Reverse(sort.IntSlice(slc)))

    midx:=len(arr)/2

    med:=slc[midx]

    return int32(med)

}

-------------------------------------------------------------------------------------------------------------