**Title**

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

**Lonely Integer**

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

**Description**

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

Given an array of integers, where all elements but one occur twice, find the unique element.

**Example**

a = [1, 2, 3, 4, 3, 2, 1]

The unique element is 4.

**Function Description**

Complete the lonelyinteger function in the editor below.

**lonelyinteger has the following parameter(s):**

int a[n]: an array of integers

**Returns**

int: the element that occurs only once

**Input Format**

The first line contains a single integer, n, the number of integers in the array.

The second line contains n space-separated integers that describe the values in a.

**Constraints**

1 <= n <= 100

It is guaranteed that n is an odd number and that there is one unique element.

0 <= a[i] <= 100, where 0 <= i <= n.

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

**Code**

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

package main

import(

    "fmt"

)

func main(){

    arr := []int32{1, 2, 3, 4, 3, 2, 1}

    var res int32

    res = lonelyinteger(arr)

    fmt.Println(res)

}

func lonelyinteger(a []int32) int32 {

    m := make(map[int32]int32)

    for \_, v := range a{

        m[v]++

    }

    var ans int32

    for k, v:= range m{

        if v ==1{

            ans = k

        }

    }

    return ans

}

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