hw_single Element in a Sorted Array

You are given a sorted array consisting of only integers where every element appears exactly twice, except for one element which appears exactly once.

Return the single element that appears only once.

Your solution must run in O(log n) time and O(1) space.

Input Format

An integer N. Array of length N.

Constraints

1 <= nums.length <= 105 0 <= nums[i] <= 105

Output Format

Print the output.

Sample Input 0

```
9
1 1 2 3 3 4 4 8 8
```

Sample Output 0

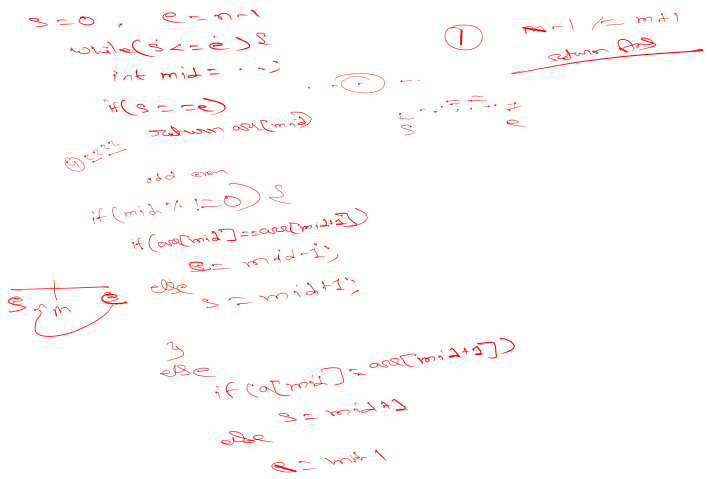
```
2
```

Sample Input 1

```
7
3 3 7 7 10 11 11
```

Sample Output 1

```
10
```



Submitted Code

Language: Java 15

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();

        System.out.print(uniqueElement(arr));
    }
    public static int uniqueElement(int [] arr){
        int s=0, e=arr.length-1;

        while(s<=e){
            int mid = s +(e-s)/2;

           if(s==e)return arr[mid];
            // if((arr[mid-1]!=arr[mid]) || (arr[mid+1]!=arr[mid])  ){
            //     return arr[mid];
            // }

           if(mid %2 ==0){
                if(arr[mid]==arr[mid+1])s= mid+2;
                else e= mid;
            }else{
                 if(arr[mid]==arr[mid+1])e= mid-1;
                 else s= mid+1;
            }
        }
        return -1;
    }
}
```

Sample Input 0

```
9
1 1 2 3 3 4 4 8 8
```

Sample Output 0

```
2
```