Bit Manipulation: Lonely Integer

Consider an array of integers where all but one of the integers occur in pairs. In other words, every element occurs exactly twice except for one unique element. Find the unique element.

For example, given the array $arr = [1, 1, 2, 3, 2]$, you would return 3.

Function Description

Complete the *findLonely* function in the editor below. It should return the unique integer in arr.

findLonely has the following parameter(s):

- *arr*: an array of integers

Input Format

The first line contains a single integer, n, denoting the number of integers in arr.
The second line contains n space-separated integers, each an element, $arr[i]$.

Constraints

- $1 \leq n < 100$
- It is guaranteed that n is an odd number.
- $0 \leq arr[i] \leq 100$, where $0 \leq i < n$.

Output Format

Print the unique number in arr on a new line.

Sample Input 0

```
1
1
```

Sample Output 0

```
1
```

Explanation 0
The array only contains a single 1, so we print 1 as our answer.

Sample Input 1

```
3
1 1 2
```

Sample Output 1

```
2
```

Explanation 1

We have two 1's and one 2. We print 2, because that's the only unique element in the array.

Sample Input 2

```
5
0 0 1 2 1
```

Sample Output 2

```
2
```

Explanation 2

We have two 0's, two 1's, and one 2. We print 2, because that's the only unique element in the array.