



Fair Rations

by [kevinsogo](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

You are the benevolent ruler of Rankhacker Castle, and today you're distributing bread to a straight line of N subjects. Each i^{th} subject (where $1 \leq i \leq N$) already has B_i loaves of bread.

Times are hard and your castle's food stocks are dwindling, so you must distribute *as few loaves as possible* according to the following rules:

1. Every time you give a loaf of bread to some person i , you must also give a loaf of bread to the person immediately in front of or behind them in the line (i.e., persons $i + 1$ or $i - 1$). In other words, you can only give a loaf of bread each to two adjacent people at a time.
2. After all the bread is distributed, each person must have an even number of loaves.

Given the number of loaves already held by each citizen, find and print the minimum number of loaves you must distribute to satisfy the two rules above. If this is not possible, print NO.

Input Format

The first line contains an integer, N , denoting the number of subjects in the bread line.

The second line contains N space-separated integers describing the respective loaves of bread already possessed by each person in the line (i.e., B_1, B_2, \dots, B_N).

Constraints

- $2 \leq N \leq 1000$
- $1 \leq B_i \leq 10$, where $1 \leq i \leq N$

Output Format

Print a single integer denoting the *minimum* number of loaves you must distribute to adjacent people in the line so that every person has an even number of loaves; if it's not possible to do this, print NO.

Sample Input 0

```
5
2 3 4 5 6
```

Sample Output 0

```
4
```

Sample Input 1

```
2
1 2
```

Sample Output 1

NO

Explanation

Sample Case 0:

The initial distribution is **(2, 3, 4, 5, 6)**. You can satisfy the problem's requirements by performing the following actions:

1. Give **1** loaf of bread each to the second and third people so that the distribution becomes **(2, 4, 5, 5, 6)**.
2. Give **1** loaf of bread each to the third and fourth people so that the distribution becomes **(2, 4, 6, 6, 6)**.

Because each of the **N** subjects now has an even number of loaves, we can stop distributing bread. We then print the total number of loaves distributed, which is **4**, on a new line.

Sample Case 1:

The initial distribution is **(1, 2)**. As there are only **2** people in the line, any time you give one person a loaf you must always give the other person a loaf. Because the first person has an odd number of loaves and the second person has an even number of loaves, no amount of distributed loaves will ever result in both subjects having an even number of loaves. Thus, we print **NO** as our answer.

[f](#) [t](#) [in](#)Submissions: [12205](#)

Max Score: 25

Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

[More](#)Current Buffer (saved locally, editable)  

Java 7



```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11         int N = in.nextInt();
12         int B[] = new int[N];
13         for(int B_i=0; B_i < N; B_i++){
14             B[B_i] = in.nextInt();
15         }
16     }
17 }
18
```

Line: 1 Col: 1

 [Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)