H   🏠 Practice   🕐 Compete   💼 Jobs   🎖 Rank   🏆 Leaderboard

Dashboard > Algorithms > Implementation > Jumping on the Clouds

Badge Progress (Details)

Points: 126 Rank: 270079

Jumping on the Clouds 🔖

by **Shafaet**

| Problem | Submissions | Leaderboard | Discussions | Editorial 🔒 |

Emma is playing a new mobile game involving n clouds numbered from 0 to $n-1$. A player initially starts out on cloud c_0, and they must jump to cloud c_{n-1}. In each step, she can jump from any cloud i to cloud $i+1$ or cloud $i+2$.

There are two types of clouds, *ordinary clouds* and *thunderclouds*. The game ends if Emma jumps onto a thundercloud, but if she reaches the last cloud (i.e., c_{n-1}), she wins the game!

Can you find the minimum number of jumps Emma must make to win the game? It is guaranteed that clouds c_0 and c_{n-1} are ordinary-clouds and it is *always possible* to win the game.

Input Format

The first line contains an integer, n (the total number of clouds).
The second line contains n space-separated binary integers describing clouds $c_0, c_1, \ldots, c_{n-1}$.

- If $c_i = 0$, the ith cloud is an ordinary cloud.

- If $c_i = 1$, the ith cloud is a thundercloud.

Constraints

- $2 \le n \le 100$

- $c_i \in \{0, 1\}$

- $c_0 = c_{n-1} = 0$

Output Format

Print the minimum number of jumps needed to win the game.

Sample Input 0

```
7
0 0 1 0 0 1 0
```

Sample Output 0

```
4
```
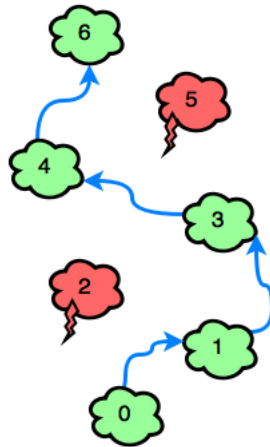
Sample Input 1

```
6
0 0 0 0 1 0
```

Sample Output 1

```
3
```

Explanation

Sample Case 0:
Because c_2 and c_5 in our input are both 1, Emma must avoid c_2 and c_5. Bearing this in mind, she can win the game with a minimum of 4 jumps:

Sample Case 1:
The only thundercloud to avoid is c_4. Emma can win the game in 3 jumps:

Submissions:<u>33886</u>

Max Score:20
Difficulty: Easy

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

Current Buffer (saved locally, editable)   ⑂ 🕒 | Java 7   ⌄ | ⤢   ⚙

```java
 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 c[] = new int[n];
13 ▾         for(int c_i=0; c_i < n; c_i++){
14 ▾             c[c_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.