Jumping on the Clouds ☆

Problem   Submissions   Leaderboard   Editorial

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

Emma is playing a new mobile game that starts with consecutively numbered clouds. Some of the clouds are thunderheads and others are cumulus. She can jump on any cumulus cloud having a number that is equal to the number of the current cloud plus 1 or 2. She must avoid the thunderheads. Determine the minimum number of jumps it will take Emma to jump from her starting postion to the last cloud. It is always possible to win the game.

For each game, Emma will get an array of clouds numbered 0 if they are safe or 1 if they must be avoided. For example, $c = [0, 1, 0, 0, 0, 1, 0]$ indexed from $0 \ldots 6$. The number on each cloud is its index in the list so she must avoid the clouds at indexes 1 and 5. She could follow the following two paths: $0 \rightarrow 2 \rightarrow 4 \rightarrow 6$ or $0 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$. The first path takes 3 jumps while the second takes 4.

Function Description

Complete the jumpingOnClouds function in the editor below. It should return the minimum number of jumps required, as an integer.

jumpingOnClouds has the following parameter(s):

- c: an array of binary integers

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 $c[i]$ where $0 \leq i < n$.

Constraints

- $2 \leq n \leq 100$
- $c[i] \in \{0, 1\}$
- $c[0] = c[n-1] = 0$

Output Format

Print the minimum number of jumps needed to win the game.
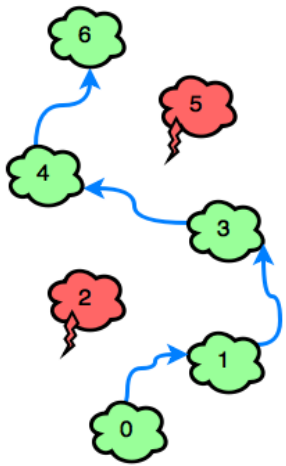
Sample Input 0

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

Sample Output 0

```
4
```

Explanation 0:

Emma must avoid $c[2]$ and $c[5]$. She can win the game with a minimum of 4 jumps:
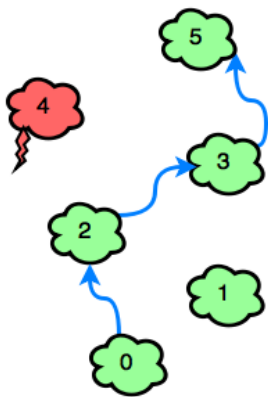
Sample Input 1

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

Sample Output 1

```
3
```

Explanation 1:

The only thundercloud to avoid is $c[4]$. Emma can win the game in 3 jumps:

Change Theme | Java 8

```java
1    import java.io.*;
2    import java.math.*;
3    import java.security.*;
4    import java.text.*;
5    import java.util.*;
6    import java.util.concurrent.*;
7    import java.util.regex.*;
8
9    public class Solution {
```

```
 10
 11        // Complete the jumpingOnClouds function below.
 12        static int jumpingOnClouds(int[] c) {
 13
 14    int jumps = 0;
 15            for(int i = 0; i < c.length; i++){
 16                if(i!=0 && c[i]==0){
 17                    jumps++;
 18                }
 19                if(i<c.length-2 && c[i+2]==0)
 20                    i++;
 21            }
 22            return jumps;
 23        }
 24
 25        private static final Scanner scanner = new Scanner(System.in);
 26
 27        public static void main(String[] args) throws IOException {
 28            BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv
        ("OUTPUT_PATH")));
 29
 30            int n = scanner.nextInt();
 31            scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
 32
```

Line: 53 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature