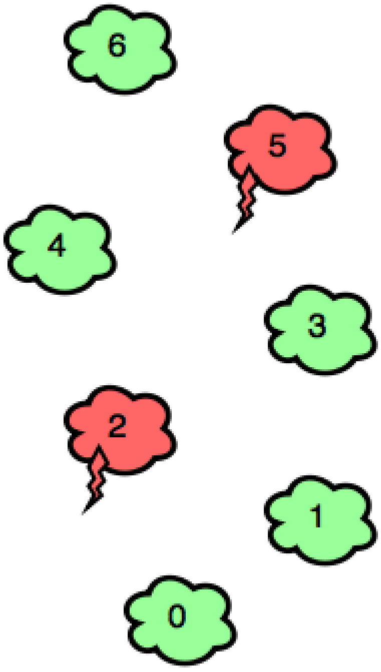
Problem: Jumping on the Clouds:

Emma is playing a new mobile game involving clouds numbered from to . A player initially starts out on cloud , and they must jump to cloud . In each step, she can jump from any cloud to cloud or cloud .

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.,), she wins the game!



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

Input Format

The first line contains an integer, (the total number of clouds).

The second line contains space-separated binary integers describing clouds .

- If , the cloud is an ordinary cloud.
- If , the cloud is a thundercloud.

Constraints

-
-
-

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
```

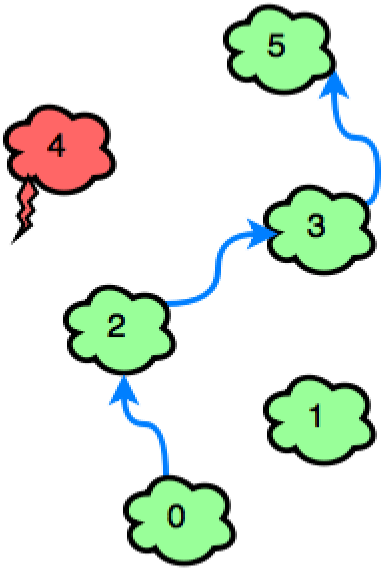
Sample Output 1

```
3
```

Explanation

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

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



Solution:

```
int main()
    {
        int clouds;
        cin>>clouds;
        int cloudType[clouds];
        for(int i=0; i<=clouds+1; i++)
            {   cin>>cloudType[i];  }



        /*Processing the data*/
        int cloud=0, i=0, jumps=0;
        while(cloud!=clouds-1)
            {
                if(cloud+2>clouds-1 || cloudType[i+2]==1) //thunderCloud
                    {
                      cloud=cloud+1;
                      jumps+=1;
                       i++;
                    }
                else   //Ordinary CLoud
                    {
                        cloud+=2;
                        jumps+=1;
                        i+=2;
                    }
            }
    cout<<jumps;

        return 0;
    }
```

- Anshul AGgarwal