Beijing was once surrounded by four rings of city walls: the Forbidden City Wall, the Imperial City Wall, the Inner City Wall, and finally the Outer City Wall. Most of these walls were demolished in the 50s and 60s to make way for roads. The walls were protected by guard towers, and there was a guard living in each tower. The wall can be considered to be a large ring, where every guard tower has exaetly two neighbors.

The guard had to keep an eye on his section of the wall all day, so he had to stay in the tower. This is a very boring job, thus it is important to keep the guards motivated. The best way to motivate a guard is to give him lots of awards. There are several different types of awards that can be given: the Distinguished Service Award, the Nicest Uniform Award, the Master Guard Award, the Superior Eyesight Award, etc. The Central Department of City Guards determined how many awards have to be given to each of the guards. An award can be given to more than one guard. However, you have to pay attention to one thing: you should not give the same award to two neighbors, since a guard cannot be proud of his award if his neighbor already has this award. The task is to write a program that determines how many different types of awards are required to keep all the guards motivated.

Input

The input contains several blocks of test eases. Each case begins with a line containing a single integer $l \leq n \leq 100000$, the number of guard towers. The next n lines correspond to the n guards: each line contains an integer, the number of awards the guard requires. Each guard requires at least 1, and at most l00000 awards. Guard i and $i+1$ are neighbors, they cannot receive the same award. The first guard and the last guard are also neighbors.

The input is terminated by a block with $n = 0$.

Output

For each test case, you have to output a line containing a single integer, the minimum number x of award types that allows us to motivate the guards. That is, if we have x types of awards, then we can give as many awards to each guard as he requires, and we can do it in such a way that the same type of award is not given to neighboring guards. A guard can receive only one award from each type.

Sample Input

```
3
4
2
2
5
2
2
2
2
2
5
1
1
1
1
1
0
```

Sample Output

```
8
5
3
```