Recover the Arrays

locked

by Shafaet

Dani is writing some arrays in her favorite text editor HackerEdit. Each line of the document describes an array in the following format:

$$e\ a_0\ a_1\ \ldots\ a_{e-1}$$

Here e is the array's number of elements and $a_0, a_1, \ldots, a_{e-1}$ are its contents.

Dani wrote m arrays in the file and left for lunch. To her dismay, her little brother Nik deleted all the newline characters from the file while she was gone! For example, consider the file in the table below:

Initial File	File With Newlines Removed
4 1 8 9 7 1 2 3 9 4 8	4 1 8 9 7 1 2 3 9 4 8

Given the contents of Dani's HackerEdit file with all the newlines removed, find the value of m (i.e., the number of arrays in the initial file).

Input Format

The first line contains an integer denoting n (the number of integers in the file).
The second line contains n space-separated integers describing each respective value in the file.

Constraints

- $2 \leq n \leq 10^5$

Output Format

Print a single integer denoting m.

Sample Input 0

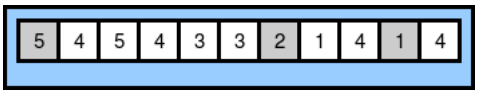
```
11
5 4 5 4 3 3 2 1 4 1 4
```

Sample Output 0

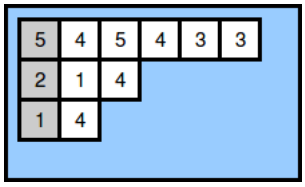
```
3
```

Explanation 0

The file looks like this after removing all the newlines:



After re-adding the newlines where they belong in the file, it looks like this:



Because there are three arrays declared in the reconstructed file, we print **3** as our answer.

Submissions: 2033
Max Score: 20
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++

```
#include <bits/stdc++.h>

using namespace std;

int main(){
    int n;
    cin >> n;
    vector<int> file(n);
    for(int file_i = 0; file_i < n; file_i++){
       cin >> file[file_i];
    }
    //  Print the number of arrays defined in 'file' to STDOUT.
    return 0;
}

```

Line: 1 Col: 1

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