Bloomberg CODECON

Memory Limit: 1350 MB
Time Limit: 5 s

Zipline Hills (250 points)

Introduction

You operate a ziplining company and are setting up a new route on an ordered series of hills. (A 'zipline route' is a set of ziplines you can ride in order without hiking from hill to hill. For example, taking a zipline from hill A to B, and then taking another zipline from hill B to C, would be a route of length two.) As everyone knows, longer zipline routes are more fun! You want to know the length of the zipline route with the most individual ziplines possible.

You know the height of each hill. Ziplines can only travel from one hill to another hill of lower height. Additionally, your ropes are only long enough to go at most two hills over. For example, from the fourth hill, you can set up a zipline going to the second, third, fifth, or sixth hills.

Given a particular order of hill heights, print out the number of separate zipline connections in the longest route that meets the above constraints.

Input Specifications

The first line of input will contain a single integer **N**, the number of hills in the test case **(1 ≤ N ≤ 100)**.

That line will be followed by **N** lines, each containing a single integer, representing the sequence of hill heights **(0 ≤ h ≤ 100)** in the test case.

Output Specifications

Based on the input, print to stdout a single integer indicating the number of hill-to-hill zipline connections in the route with the most possible such connections.

Sample Input/Output

Input

```
6
80
24
36
53
91
17
```

Output

```
3
```

Explanation

The longest route possible has three zipline connections: 91 → 53 → 36 → 24.

Input

```
6
67
40
21
9
37
77
```

Output

```
3
```

Explanation

Two different routes have length three: 77 → 37 → 21 → 9, and 67 → 40 → 21 → 9.

Note that the longer chain 67 → 40 → 37 → 21 → 9 is **not** a solution, because 40 → 37, three hills apart, is farther than the maximum allowed gap of two hills.