

Song Thief

Mr.X wants to be a musician but has zero talent. But to his luck he has come across an old long symphony his great grandfather has composed. He wants to make as many good songs as he could with this.

The symphony is **N** minutes long and each minute of the symphony has a different pitch, which is a positive integer value. A part of symphony equal to or longer than **3 minutes** is considered melodious if,

1. The pitch of the first and last minute are **strictly greater** than the pitches of other minutes, or
2. The pitch of the first and last minute are **strictly lower** than the pitches of other minutes.

For example, **(2 3 4 1)** and **(6 1 1 2)** are melodious, **(3 6 3 1)** and **(2 4 1 6)** are not.

A song is good if there is at least one melodious parts in it. So X wants to cut the symphony in to smaller continuous sections (of any length) of good songs. You need to find the maximum number of songs X could make. Note that two songs cannot use the same melody

For example, he could make two songs **(2 3 1 5)** and **(6 4 3 5)** out of symphony **(2 3 1 5 6 4 3 5)**. Both of these songs are good because they both contain melodious parts, **(2 3 1)** and **(6 4 3 5)** respectively.

The symphony is good, that is it has at least one melodious part in it.

Input Format

First line contains a single integer **N**, the length of the symphony.
The next line contains **N** positive integers separated by space representing the pitches of each minute.

Constraints

- $5 \leq N \leq 10^3$
- $1 \leq \text{pitch} \leq 10^6$

Limits

- **Time Limit:** 1s
- **Memory Limit:** 256MB

Output Format

A single number containing **the number of good songs X can make**

Sample Input 0

8
2 3 1 5 6 4 3 5

Sample Output 0

2

Explanation 0

Explained above