Question 1

Problem: Longest Consecutive Increasing Subsequence (Incline)

A consecutive increasing subsequence of an array is a sequence of numbers where each number is strictly greater than the previous number and all numbers appear consecutively in the array.

For example, in the array:

[2, 3, 1, 1, 5, 6]

The consecutive increasing subsequences are:

- [2, 3] → length 2
- [1, 5, 6] → length 3

The longest one has length 3.

Write a program that reads an array of integers and finds the length of the longest consecutive increasing subsequence.

Input

- The first line contains a single integer n — the length of the array.
- The second line contains n space-separated integers — the elements of the array.

Output

- A single integer — the length of the longest consecutive increasing subsequence.

Example 1

Input

```
6

2 3 1 1 5 6
```

Output

```
3
```

Explanation:
The consecutive increasing subsequences are:

- [2,3] → length 2
- [1,5,6] → length 3 ✅ (longest)

Example 2

Input

7

2 3 1 11 5 5 6

Output

2

Explanation:
The consecutive increasing subsequences are:

- **[2,3] → length 2 ✅ (longest starting at index 0)**
- **[1,11] → length 2**
- **[5,6] → length 2**

All other sequences are shorter. The longest length = 2.

Detailed Walkthrough

1. **Start at the first number and compare it with the next:**
    - **If numbers[i] < numbers[i+1], increase the current run counter (count).**
    - **Otherwise, the sequence ended; check if this run is the longest (highest) and reset count.**
2. **After finishing the array, make sure to check the last run in case the array ends with an increasing sequence.**
3. **Add 1 to the count when reporting the length because count counts increments, not the number of elements.**

Notes for Students

- **Only consider strictly increasing consecutive elements.**
- **A single number alone counts as a subsequence of length 1.**
- **Be careful to check the last sequence after looping.**
- **This is different from the general Longest Increasing Subsequence (LIS), where numbers can be skipped.**

Question 2

Problem: Longest Increasing Subsequence

A subsequence of an array is a sequence of numbers that can be derived from the array by deleting some or no elements **without changing the order of the remaining elements**.

For example, if the array is:

[2, 3, 1, 11, 5, 5, 6]

Some possible increasing subsequences are:

- [2, 3, 11]
- [2, 3, 5, 6]
- [1, 5, 6]

The **length** of a subsequence is the number of elements in it.

Write a program that reads an array of integers and **finds the length of the longest increasing subsequence (LIS)**.

Input

- The first line contains a single integer n — the length of the array.
- The second line contains n space-separated integers — the elements of the array.

Output

- A single integer — the length of the longest increasing subsequence.

Example 1

Input

6

2 3 1 1 5 6

Output

4

Explanation:
The longest increasing subsequence is [2, 3, 5, 6], which has length 4.

Example 2

Input

7

2 3 1 11 5 5 6

Output

4

Explanation:
The longest increasing subsequence is [2, 3, 5, 6] (or [2, 3, 5, 6] via different indices). Length = 4.

Detailed Subset Explanation

To understand **why the LIS is 4**, consider **all possible increasing subsequences**:

1. Start with 2:
   - [2, 3] → [2, 3, 5] → [2, 3, 5, 6] ✅
   - [2, 3, 11] → ends at 3 elements, shorter than 4.
2. Start with 3:
   - [3, 5] → [3, 5, 6] → length 3 (not the longest).
3. Start with 1:
   - [1, 5] → [1, 5, 6] → length 3 (not the longest).
4. Start with 11:
   - Only [11] → length 1.
5. Start with 5:
   - [5, 6] → length 2.
6. Start with last 5:
   - [5, 6] → length 2.
7. Start with 6:
   - [6] → length 1.

✅ So the **longest increasing subsequence** is [2, 3, 5, 6] → **length = 4**.

Notes for Students

- A subsequence **does not need to be contiguous** in the array.
- If there are multiple subsequences with the same maximum length, **only the length matters**, not which subsequence you pick.
- The **dynamic programming approach** is recommended for arrays up to 1000 elements.