A. Kefa and First Steps

time limit per test

2 seconds

memory limit per test

256 megabytes

input

standard input

output

standard output

Kefa decided to make some money doing business on the Internet for exactly *n* days. He knows that on the *i*-th day (1 ≤ *i* ≤ *n*) he makes *ai* money. Kefa loves progress, that's why he wants to know the length of the maximum non-decreasing subsegment in sequence *ai*. Let us remind you that the subsegment of the sequence is its continuous fragment. A subsegment of numbers is called non-decreasing if all numbers in it follow in the non-decreasing order.

Help Kefa cope with this task!

**Input**

The first line contains integer *n* (1 ≤ *n* ≤ 105).

The second line contains *n* integers *a*1,  *a*2,  ...,  *an* (1 ≤ *ai* ≤ 109).

**Output**

Print a single integer — the length of the maximum non-decreasing subsegment of sequence *a*.

**Sample test(s)**

**input**

6  
2 2 1 3 4 1

**output**

3

**input**

3  
2 2 9

**output**

3

**Note**

In the first test the maximum non-decreasing subsegment is the numbers from the third to the fifth one.

In the second test the maximum non-decreasing subsegment is the numbers from the first to the third one.

<http://codeforces.com/problemset/problem/580/A>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

string[] array = Console.ReadLine().Split(' ');

int[] arr = new int[array.Length];

for (int j = 0; j < arr.Length; j++)

{

arr[j] = int.Parse(array[j]);

}

int longestLen = 0;

int i = 0;

while (i < arr.Length)

{

int len = 1;

while (i + 1 < arr.Length && arr[i] <= arr[i + 1])

{

len++;

i++;

}

longestLen = Math.Max(len, longestLen);

i++;

}

Console.WriteLine(longestLen);

//for (int i = 0; i < arr.Length; i++)

//{

// Console.WriteLine(arr[i]);

//}

Console.ReadLine();

}

}

}