G. Ни больше ни меньше

|  |  |
| --- | --- |
| Ограничение времени | 2 секунды |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Дан массив целых положительных чисел *a* длины *n*. Разбейте его на **минимально возможное** количество отрезков, чтобы каждое число было не меньше длины отрезка которому оно принадлежит. Длиной отрезка считается количество чисел в нем.

Разбиение массива на отрезки считается корректным, если каждый элемент принадлежит ровно одному отрезку.

Формат ввода

Первая строка содержит одно целое число *t* *(1 ≤ t ≤ 1 000)* — количество наборов тестовых данных. Затем следуют *t* наборов тестовых данных.

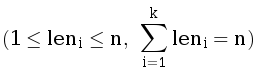
Первая строка набора тестовых данных содержит одно целое число *n* *(1 ≤ n ≤ 105)* — длину массива.

Следующая строка содержит *n* целых чисел *a1, a2, …, an* *(1 ≤ ai ≤ n)* — массив *a*.

Гарантируется, что сумма *n* по всем наборам тестовых данных не превосходит *2 ⋅ 105*.

Формат вывода

Для каждого набора тестовых данных в первой строке выведите число *k* — количество отрезков в вашем разбиении.

Затем в следующей строке выведите *k* чисел *len1, len2, …, lenk*  — длины отрезков в порядке слева направо.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  5  1 3 3 3 2  16  1 9 8 7 6 7 8 9 9 9 9 9 9 9 9 9  7  7 2 3 4 3 2 7 | 3  1 2 2  3  1 6 9  3  2 3 2 |

Примечания

Ответы в примере соответствуют разбиениям:

*{[1], [3, 3], [3, 2]}*

*{[1], [9, 8, 7, 6, 7, 8], [9, 9, 9, 9, 9, 9, 9, 9, 9]}*

*{[7, 2], [3, 4, 3], [2, 7]}*

В первом наборе тестовых данных набор длин *{1, 3, 1}*, соответствующий разбиению *{[1], [3, 3, 3], [2]}*, также был бы корректным.

using System;

class Program

{

static void Main()

{

int N = Convert.ToInt32(Console.ReadLine());

for (int u = 0; u < N; u++)

{

int n = Convert.ToInt32(Console.ReadLine());

string[] input1 = Console.ReadLine().Split();

int[] array = new int[n];

for (int i = 0; i < n; i++)

{

array[i] = int.Parse(input1[i]);

}

bool flag = true;

int result = 0;

int[] resArr = new int[n];

int index = 0;

int summ = 0;

int len = array[0];

while (flag)

{

if (array[index] < len) len = array[index];

if (summ + 1 <= len)

{

resArr[result]++;

summ++;

}

else

{

summ = 1;

len = array[index];

result++;

resArr[result]++;

}

index++;

if (index == n) flag = false;

}

result++;

Console.WriteLine(result);

for (int i = 0; i < result; i++)

{

Console.Write(resArr[i] + " ");

}

Console.WriteLine();

}

}

}