

## Очередь на Луну. 7-10 классы

Коротышки из Цветочного города готовятся отправиться в путешествие на Луну. Знайка поручил Незнайке, выстроить всех коротышек в очередь на посадку в космический корабль так, чтобы они были упорядочены по росту: первым должен стоять самый низкий, последним – самый высокий. Однако, пока Незайка витал в облаках, все перепутались и встали в очередь, кто как хотел.

Строгий Знайка решил навести порядок и наказать всех тех, кто встал в очередь не по порядку. Знайка решил выбрать самую длинную цепочку коротышек, которые уже стоят (не обязательно рядом) в правильном порядке (по неубыванию роста). Всех остальных, кто не входит в эту цепочку, он будет наказывать, заставляя учить лунные правила. Помогите Знайке узнать, каково количество коротышек, которые окажутся вне цепочки и будут наказаны.

Требуется составить программу, которая считывает натуральное число  $N$ :  $0 < N < 5001$  – общее количество коротышек в очереди, а затем  $N$  натуральных чисел  $H[i]$ :  $0 < H[i] < 1000000001$ .  $H[i]$  – рост коротышки, стоящего на  $i$ -ом месте в очереди. Программа находит наименьшее целое число  $K$  ( $-1 < K < N$ ) – количество коротышек, которые не входят в цепочку из коротышек, стоящих в очереди на местах с номерами  $i_1, i_2, \dots, i_{N-K}$ , где  $i_1 < i_2 < \dots < i_{N-K}$ , и  $H[i_1], H[i_2], \dots, H[i_{N-K}]$  является неубывающей, т. е.  $H[i_j] < H[i_{j+1}]$  или  $H[i_j] = H[i_{j+1}]$  для любого  $j$  ( $0 < j < N - K$ ).

**Формат ввода:** В первой строке задано число  $N$ :  $0 < N < 5001$ . Во второй строке заданы  $N$  натуральных чисел  $H[i]$ :  $0 < H[i] < 1000000001$ .

**Формат вывода:** Выводится искомое число  $K$ :  $-1 < K < N$ .

Ввод примера №1:		Ввод примера №2:		Ввод примера №3:
6		5		5
3 29 5 5 28 6		5 8 10 4 1		1 1 2 2 2
Вывод примера №1:		Вывод примера №2:		Вывод примера №3:
2		2		0
Пояснение:		Пояснение:		Пояснение:
Цепочка: 3 5 5 6		Цепочка: 5 8 10		Все выстроились верно.
Вне цепочки: 29 28		Вне цепочки: 4 1		

## Пояснения к решению

Вместо того чтобы искать наименьшее количество наказуемых, будем искать наибольшее количество тех, кого наказывать не нужно. Тогда ответ можно получить как  $ans = N - LnDS$ , где  $LnDS$  – это длина наибольшей невозрастающей подпоследовательности. При данных ограничениях на  $N \leq 10^4$  задачу можно решить с помощью метода динамического программирования. Для этого определим «динамику»  $dp[i]$  как длину наибольшей неубывающей подпоследовательности, которая заканчивается на  $H[i]$  элементе. Легко видеть, что  $dp[0]$  можно положить за 1. Пересчет можно выполнять в цикле по следующему соотношению  $dp[i] = \max(dp[i], dp[j] + 1)$  для всех  $j < i$  с условием, что  $H[j] \leq H[i]$ . 4. Учитывая, что наибольшая неубывающая подпоследовательность закончится на каком-то элементе,  $LnDS = \max(dp[0], dp[1], \dots, dp[N - 1])$

## Код возможного решения

```
#include <bits/stdc++.h>

using namespace std;

int main(void) {
    int n;
    cin >> n;
    vector<int> arr(n);
    for (auto &i : arr) cin >> i;

    if (n == 0)
    {
        cout << 0 << endl;
    }
}
```

```

        return 0;
    }

    vector<int> dp(n, 1);

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < i; ++j) {
            if (arr[j] <= arr[i] && dp[j] + 1 > dp[i]) {
                dp[i] = dp[j] + 1;
            }
        }
    }

    int len = *max_element(dp.begin(), dp.end());
    int punish = n - len;
    cout << punish << endl;

    return 0;
}

```