

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

## Алгоритмы и структуры данных

Отчёт по лабораторной работе №9 (1604)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

## 1. Цель работы

Решить задачу №1604 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1604>

## 2. Задача

### Условие

Главный бульдог-полицейский Страны Дураков решил ввести ограничение скоростного режима на автомобильной трассе, ведущей от Поля Чудес к пруду Черепахи Тортиллы. Для этого он заказал у Папы Карло  $n$  знаков ограничения скорости. Папа Карло слабо разбирался в дорожном движении и поэтому изготовил знаки с разными ограничениями на скорость: 49 км/ч, 34 км/ч, 42 км/ч, и т.д. Всего получилось  $k$  различных ограничений:  $n_1$  знаков с одним ограничением,  $n_2$  знаков со вторым ограничением, и т.д. ( $n_1 + \dots + n_k = n$ ). Бульдог-полицейский ничуть не расстроился, получив такие знаки, напротив, он решил извлечь из этого экономическую выгоду. Дело в том, что по Правилам дорожного движения Страны Дураков ограничение на скорость действует вплоть до следующего знака. Если на знаке написано число 60, это означает, что участок от данного знака до следующего нужно проехать ровно со скоростью 60 километров в час — не больше и не меньше. Бульдог распорядился расставить знаки так, чтобы обогатившимся на Поле Чудес автолюбителям во время своего движения по трассе приходилось как можно больше раз менять скорость. Для этого нужно расставить имеющиеся знаки в правильном порядке. Если Вы поможете бульдогу это сделать, то он готов будет поделиться с Вами частью своих доходов.

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

### Исходные данные

В первой строке дано число  $k$  — количество различных типов знаков с ограничением скорости ( $1 \leq k \leq 10000$ ). Во второй строке через пробел перечислены целые положительные числа  $n_1, \dots, n_k$ . Сумма всех  $n_i$  не превосходит 10000.

### Результат

Выведите  $n$  целых чисел в пределах от 1 до  $k$  — порядок, в котором нужно расставить по трассе имеющиеся знаки. Вне зависимости от того, какой знак стоит первым, считается, что, проезжая его, водитель меняет скорость, так как до этого ограничения не действовали. Если задача имеет несколько решений, выведите любое.

### Пример

Исходные данные	Результат
2 2 2	1 2 1 2

### 3. Материалы работы

#### 3.1. Объяснение алгоритма

Для выполнения задачи необходимо минимизировать количество стоящих рядом одинаковых знаков. Для этого в начале по очереди в массив записываются количество каждого из типов знаков, после чего происходит расстановка. В цикле следующий знак определяется как знак, количество которых максимально и тип следующего знака отличен от предыдущего. После установки знака, от убирается из массива (количество знаков этого типа уменьшается на 1).

#### 3.2. Код программы.

```
1. #include <iostream>
2.
3. using namespace std;
4.
5. int main(){
6.
7.     int k, sum = 0;
8.     cin >> k;
9.     int array[k];
10.    for (int i = 0; i < k; i++) {
11.        cin >> array[i];
12.        sum += array[i];
13.    }
14.    int last_index = -1;
15.    int max_temp_index;
16.    for (int i = 0; i < sum; i++){
17.        max_temp_index = 0;
18.        if(last_index == 0) max_temp_index = 1;
19.        for (int j = 1; j < k; j++){
20.            if(last_index != j && array[j] >= array[max_temp_index])
21.                max_temp_index = j;
22.        }
23.        if (array[max_temp_index] == 0)
24.            max_temp_index = last_index;
25.        array[max_temp_index] -= 1;
26.        last_index = max_temp_index;
27.        cout << max_temp_index + 1 << ' ';
28.    }
29. }
```

### 4. Результат выполнения и ссылка на репозиторий GitHub

Автор: [Daniil Marukhlenko](#) • Задача: [В Стране Дураков](#)

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">9249680</a>	18:30:21 1 мар 2021	<a href="#">Daniil Marukhlenko</a>	<a href="#">1604</a>	G++ 9.2 x64	Accepted		0.078	432 КБ

[https://github.com/japersik/algorithms\\_and\\_data\\_structures/](https://github.com/japersik/algorithms_and_data_structures/)



## 5. Вывод

Работа выполнена, задача решена, полученный алгоритм соответствует требованиям проверки программы.