

# Стажировка весна-лето 2023 | бэкенд

12 фев 2023, 16:51:09

старт: 12 фев 2023, 16:02:11

финиш: 12 фев 2023, 21:02:11

до финиша: 04:10:59

начало: 1 фев 2023, 00:00:00

длительность: 05:00:00

## А. Лучше клавиатуры не сыскать?

	Все языки	GNU C++20 10.2	Clang14 C++20
Ограничение времени	3 секунды	1 секунда	1 секунда
Ограничение памяти	256Mb	256Mb	256Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

**Объявление:** если у вас есть **жалобы / претензии / вопросы** насчет задач, то советуем для начала ознакомиться с [расширенной инструкцией](#), содержащей в том числе ответы на самые частые вопросы.

**Начало условия:** Кузя снова не успел сдать реферат по языковым разнообразиям в срок. «Наверное, у меня какая-то неоптимальная клавиатура...» — подумал Кузя и решил изобрести самую оптимальную клавиатуру для набора одним пальцем.

Кузя решил, что его клавиатура будет содержать  $N$  рядов с клавишами (разные ряды могут содержать разные количества клавиш). Все клавиши на клавиатуре будут уникальными.

Чтобы оценить качество клавиатуры для набора определённого текста, Кузя ввёл понятие «разнорядности»:

- Пусть текст представляет из себя строку  $S$  длины  $K$ ;
- Назовём переход между **соседними в тексте** символами  $S_{i-1}$  и  $S_i$  ( $2 \leq i \leq K$ ) разнорядным, если данные символы находятся на различных рядах клавиатуры;
- Разнорядность равна количеству разнорядных переходов при наборе текста на заданной клавиатуре.

Например, пусть текст  $S$  равен  $ABCAD$ , а клавиатура содержит два ряда клавиш  $AC$  и  $BD$ . В таком случае при наборе текста будет ровно 3 разнорядных перехода:

- $S_1 = A$  в  $S_2 = B$  (ряд 1 в ряд 2);
- $S_2 = B$  в  $S_3 = C$  (ряд 2 в ряд 1);
- $S_4 = A$  в  $S_5 = D$  (ряд 1 в ряд 2).

Вас, как лучшего игрока в Тетрис среди знакомых, Кузя просит вычислить разнорядность созданной им клавиатуры на последнем из Кузиных рефератов.

## Формат ввода

Первая строка содержит одно целое число  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ) — количество клавиш на клавиатуре.

Вторая строка содержит  $N$  целых чисел  $c_i$  ( $0 \leq c_i \leq 10^9$ ) — идентификаторы символов на клавишах. Гарантируется, что все значения  $c_i$  различны.

Третья строка содержит  $N$  целых чисел  $r_i$  ( $1 \leq r_i \leq 10^9$ ). Число  $r_i$  задает номер ряда на клавиатуре, в котором расположена клавиша с символом  $c_i$ .

Четвертая строка содержит одно целое число  $K$  ( $1 \leq K \leq 2 \cdot 10^5$ ) — количество символов в реферате.

Пятая строка содержит  $K$  целых чисел  $s_j$  ( $0 \leq s_j \leq 10^9$ ) — идентификаторы символов реферата в порядке набора на клавиатуре.

Гарантируется, что для любого  $s_j$  существует такой  $i$ , что  $s_j = c_i$  — любой символ из реферата присутствует на клавиатуре.

## Формат вывода

Выведите единственное целое число — разнорядность заданной во входных данных конструкции клавиатуры на реферате  $S$ .

## Пример 1

Ввод

Вывод

4  
1 2 3 4  
1 2 1 2  
5  
1 2 3 1 4

3

## Пример 2

Ввод

Вывод

3  
42 3 14  
1 3 3  
4  
3 14 14 3

0

## Примечания

Пояснение к **первому** тестовому примеру.

Данный пример соответствует примеру из условия, где каждой букве сопоставлен её номер в латинском алфавите:

- «A» — 1;
- «B» — 2;
- «C» — 3;
- «D» — 4.

Как и в условии, первый ряд содержит символы 1 и 3, второй — 2 и 4.

Ответ 3 получается аналогично условию:

- при переходе от символа 1 к символу 2;
- при переходе от символа 2 к символу 3;
- при переходе от символа 1 к символу 4.

Пояснение ко **второму** тестовому примеру.

1. Первый ряд содержит клавишу с идентификатором 42;
2. Второй ряд пустует;
3. Третий ряд — клавиши с идентификаторами 3 и 14.

Все клавиши, используемые при наборе текста реферата, находятся в 3-м ряду, поэтому разнорядность клавиатуры равна 0.

Язык

GNU C++20 10.2

Набрать здесь

Отправить файл

```

1 #include<iostream>
2 #include<vector>
3 #include<unordered_map>
4 #include<math.h>
5 using namespace std;
6
7 int main() {
8     //ios::sync_with_stdio(false);
9     //cin.tie(nullptr);
10
11     int n;
12     cin >> n;
13     vector<int> c(n);
14     for (size_t i = 0; i < n; i++) cin >> c[i];
15
16     unordered_map<int, int> m;
17     int tmp;
18     for (size_t i = 0; i < n; i++)
19     {
20         cin >> tmp;
21         m.insert({ c[i], tmp });
22     }
23
24     int k;
25     cin >> k;
26     vector<int> s(k);
27     for (size_t i = 0; i < k; i++) cin >> s[i];
28
29     int sum = 0;
30     for (size_t i = 1; i < k; i++)
31     {
32         if (m[s[i]] != m[s[i - 1]]) sum += 1;
33     }
34     cout << sum;
35 }
36

```

Отправить

Следующая