

### 3. Коллекционер Диего

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

Диего увлекается коллекционированием наклеек. На каждой из них написано число, и каждый коллекционер мечтает собрать наклейки со всеми встречающимися числами.

Диего собрал  $N$  наклеек, некоторые из которых, возможно, совпадают. Как-то раз к нему пришли  $K$  коллекционеров.  $i$ -й из них собрал все наклейки с номерами не меньшими, чем  $p_i$ . Напишите программу, которая поможет каждому из коллекционеров определить, сколько недостающих ему наклеек есть у Диего. Разумеется, гостей Диего не интересуют повторные экземпляры наклеек.

#### Формат ввода

В первой строке содержится единственное число  $N$  ( $0 \leq N \leq 100\,000$ ) — количество наклеек у Диего.

В следующей строке содержатся  $N$  целых неотрицательных чисел (не обязательно различных) — номера наклеек Диего. Все номера наклеек не превосходят  $10^9$ .

В следующей строке содержится число  $K$  ( $0 \leq K \leq 100\,000$ ) — количество коллекционеров, пришедших к Диего. В следующей строке содержатся  $K$  целых чисел  $p_i$  ( $0 \leq p_i \leq 10^9$ ), где  $p_i$  — наименьший номер наклейки, не интересующий  $i$ -го коллекционера.

#### Формат вывода

Для каждого коллекционера в отдельной строке выведите количество **различных** чисел на наклейках, которые есть у Диего, но нет у этого коллекционера.

#### Пример 1

| Ввод | Вывод |
|------|-------|
| 1    | 0     |
| 5    | 1     |
| 2    |       |
| 4 6  |       |

#### Пример 2

| Ввод     | Вывод |
|----------|-------|
| 3        | 3     |
| 100 1 50 | 0     |
| 3        | 2     |
| 300 0 75 |       |

Язык 

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <algorithm>
2 #include <iostream>
3 #include <set>
4 #include <vector>
5
6 int main() {
7     int countStickers;
8     std::cin >> countStickers;
9
10    std::set<int> stickers;
11    for (int tmp, i = 0; i < countStickers; ++i) {
12        std::cin >> tmp;
13        stickers.insert(tmp);
14    }
```

```
15     '  
16     std::vector<int> optimization_time(stickers.begin(), stickers.end());  
17  
18     int countCollectors;  
19     std::cin >> countCollectors;  
20  
21     std::vector<int> minSticker;  
22     for (int i = 0, tmp; i < countCollectors ; ++i) {  
23         std::cin >> tmp;  
24         minSticker.push_back(tmp);  
25     }  
26  
27     for (auto it = minSticker.begin(), ite = minSticker.end(); it != ite; ++it) {  
28         std::cout << std::distance(optimization_time.begin(), std::lower_bound(optimization_time.begin(), optimization_time  
29     }  
30     return 0;  
31 }
```

[Отправить](#)[Предыдущая](#)[Следующая](#)