Оглавление

[A. Новая история 1](#_Toc148034187)

[Формат ввода 2](#_Toc148034188)

[Формат вывода 2](#_Toc148034189)

[**Пример 1** 3](#_Toc148034190)

[**Пример 2** 3](#_Toc148034191)

[Примечания 3](#_Toc148034192)

[B. Разнообразие 4](#_Toc148034193)

[Формат ввода 4](#_Toc148034194)

[Формат вывода 5](#_Toc148034195)

[**Пример 1** 5](#_Toc148034196)

[**Пример 2** 5](#_Toc148034197)

[**Пример 3** 6](#_Toc148034198)

[Примечания 7](#_Toc148034199)

[C. Запрос к таблице 8](#_Toc148034200)

[Формат ввода 8](#_Toc148034201)

[Формат вывода 8](#_Toc148034202)

[Пример 8](#_Toc148034203)

[Примечания 9](#_Toc148034204)

[D. Межпланетная организация 10](#_Toc148034205)

[Формат ввода 11](#_Toc148034206)

[Формат вывода 11](#_Toc148034207)

[**Пример 1** 12](#_Toc148034208)

[**Пример 2** 12](#_Toc148034209)

[Примечания 12](#_Toc148034210)

[E. Близость 14](#_Toc148034211)

[Формат ввода 14](#_Toc148034212)

[Формат вывода 14](#_Toc148034213)

[**Пример 1** 15](#_Toc148034214)

[**Пример 2** 15](#_Toc148034215)

[Примечания 15](#_Toc148034216)

# 

# A. Новая история

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Сейчас активно развивается новая история, основателем которой является Профессор А.С. Багиров. Он выяснил, что на протяжении многих лет на земле вместе с людьми существовали ящеры. Строительство пирамид, захват Байкала и еще много разных событий произошли благодаря ящерам.

Учёные ещё не выяснили, сколько времени ящеры существовали на земле. Они находят разные данные в виде даты начала и даты окончания, и чтобы проверить их на корректность, необходимо посчитать, сколько дней ящеры существовали для двух конкретных дат. Календарь ящеров очень похож на григорианский, лишь с тем исключением, что там нет високосных годов.

Вам даны дата начала и дата окончания существования ящеров, нужно найти количество полных дней и секунд в неполном дне, чтобы учёные смогли оценить, насколько даты корректны.

## Формат ввода

В первой строке содержатся 6 целых чисел year1,month1,day1,hour1,min1,sec1(1≤year1≤9999,1≤month1≤12,1≤day1≤31,0≤hour1≤23,0≤min1≤59,0≤sec1≤59)- дата начала существования ящеров.

Во второй строке содержатся 6 целых чисел year2,month2,day2,hour2,min2,sec2(1≤year2≤9999,1≤month2≤12,1≤day2≤31,0≤hour2≤23,0≤min2≤59,0≤sec2≤59)- дата окончания существования ящеров.

Гарантируется, что дата начала меньше,чем дата конца.

## Формат вывода

В первой и единственной строке выведите 2 числа: количество дней, сколько существовали ящеры, а также количество секунд в неполном дне.

### **Пример 1**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 980 2 12 10 30 1  980 3 1 10 31 37 | 17 96 |

### **Пример 2**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 1001 5 20 14 15 16  9009 9 11 12 21 11 | 2923033 79555 |

## Примечания

В одном году 365 дней. Год делится на 12 месяцев, количество дней в каждом месяце: [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]. В одном дне 24 часа (от 0 до 23), в одном часу 60 минут (от 0 до 59), в одной минуте 60 секунд (от 0 до 59).

В первом тестовом примере года совпадают, а между 12 февраля и 1 марта прошло 17 полных дней, начало было в 10:30:01, а конец в 10:31:37,таким образом прошла 1 минута и 36 секунд, что в сумме получается 96 секунд.

# B. Разнообразие

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

Два друга A и B постоянно играют в коллекционную карточную игру (ККИ), поэтому у каждого игрока скопилась довольно большая коллекция карт.

Каждая карта в данной игре задаётся целым числом (одинаковые карты — одинаковыми числами, разные карты — разными).

Таким образом коллекцию можно представить как неупорядоченный набор целых чисел (с возможными повторениями).

После каждого изменения коллекций друзья вычисляют показатель **разнообразия**следующим образом:

* A и B выкладывают на стол все карты из своей коллекции в два раздельных ряда;
* Далее друзья итеративно делают следующее:
  1. Если среди лежащих на столе карт игрока A есть такая же карта, как и среди лежащих карт игрока B — каждый игрок убирает данную карту со стола;
  2. Если таковых совпадений нет — процесс заканчивается.
* **Разнообразием**коллекций друзья называют суммарное количество оставшихся карт на столе.

Даны начальные состояния коллекций игроков, а также Q изменений их коллекций. После каждого изменения необходимо вычислить **разнообразие**коллекций друзей.

## Формат ввода

В первой строке через пробел заданы числа N, M, Q(1≤N,M,Q≤105) — количество карт в коллекциях игрока A и B и количество изменений соответственно.

Вторая строка содержит через пробел N целых чисел ai(1≤ai≤109) — карты в коллекции игрока A.

Третья строка содержит через пробел M целых чисел bj(1≤bj≤109) — карты в коллекции игрока B.

Далее на каждой из следующих Q строк описано изменение коллекции: через пробел заданы typekplayerkcardk(typek=±1;playerk∈(A,B);1≤cardk≤109) — тип k-го изменения, имя игрока и значение карты:

* Если type=1, то в коллекцию игрока player добавился экземпляр карты card;
* Если type=−1, то из коллекции игрока player удалился **один**экземпляр карты card.
* Гарантируется, что при запросе type=−1 хотя бы один экземпляр карты card присутствует в коллекции игрока player.

## Формат вывода

Необходимо вывести через пробел Q целых чисел — разнообразие коллекций игроков A и B после k-го изменения.

### **Пример 1**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2 5 10  1 2  1 2 3 4 5  1 A 3  1 A 4  1 A 5  1 A 6  1 A 7  -1 A 1  1 B 7  -1 A 6  -1 B 1  1 A 7 | 2 1 0 1 2 3 2 1 0 1 |

### **Пример 2**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3 3 5  1000 2000 1001  1001 2001 1000  1 A 100000  -1 B 2001  1 B 2000  1 B 100001  1 A 1 | 3 2 1 2 3 |

### **Пример 3**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3 3 20  1 6 7  2 4 5  1 A 2  1 B 1  1 B 8  1 B 5  1 A 3  1 A 2  1 B 10  1 A 9  1 A 8  1 B 7  -1 A 1  -1 B 5  -1 B 5  -1 B 4  -1 A 6  -1 A 8  -1 A 2  -1 B 8  -1 B 10  -1 A 2 | 5 4 5 6 7 8 9 10 9 8 9 8 7 6 5 6 5 4 3 4 |

## Примечания

В **первом примере**после первого изменения коллекции следующие: 1,2,3 и 1,2,3,4,5, разнообразие - 2, т.к. игроки уберут общие карты 1,2,3, оставив 4,5.

Далее поочередно первому игроку добавляются 4,5, сокращая различие между коллекциями игроков на 1 в обоих случаях.

Четвёртое и пятое изменение добавляет первому игроку карты 6,7, которых нет у игрока B, поэтому разнообразие увеличивается на 1 в обоих случаях.

Шестое изменение увеличивает разнообразие, т.к. до него 1 была общей картой игроков.

Седьмое уменьшает добавлением общей 7-рки, восьмое уменьшает убиранием из коллекции A 6-рки, которой нет у B.

Девятое изменение приводит коллекции к одинаковому виду - разнообразие 0.

Последнее изменение добавляет ещё одну 7 игроку A, увеличивая разнообразие, т.к. у второго игрока 7 лишь одна.

Во **втором примере**на первом шаге коллекции 1000,2000,1001,100000 ; 1001,2001,1000 - на столе останутся 2000,100000 и 2001.

На втором шаге разнообразность понижается на 1, т.к. 2001 была только в коллекции второго игрока.

На третьем шаге второму игроку добавляется карта 2000, которая есть у первого игрока, поэтому разнообразие уменьшается до 1.

На четвёртом шаге разнообразие повышается при добавлении новой карты 100001 игроку B.

На пятом шаге разнообразие увеличивается, т.к. у второго игрока карт 1 нет.

# C. Запрос к таблице

|  | Все языки | Python 3.9 (PyPy 7.3.11) | Python 3.11.4 |
| --- | --- | --- | --- |
| Ограничение времени | 2 секунды | 4 секунды | 4 секунды |
| Ограничение памяти | 512Mb | 512Mb | 512Mb |
| Ввод | стандартный ввод или input.txt | | |
| Вывод | стандартный вывод или output.txt | | |

Петя пришел на стажировку в Яндекс, и первая его задача была познакомиться с SQL.

У Пети есть табличка, состоящая из N строк и M столбцов, значениями которой являются целые числа. Каждой колонке соответствует уникальное имя — строка из латинских символов.

Пете заданы Q ограничений вида: ColumnNamekqkvalk.

qk может принимать два значения:

1. > — учитывать только те строки, где значения в ColumnNamek строго больше valk;
2. < — учитывать только те строки, где значения в ColumnNamek строго меньше valk.

Задача Пети заключается в том, чтоб посчитать **сумму**во всех строках, которые удовлетворяют **всем ограничениям**. Юный стажер уже написал скрипт и вычислил ответ. Но Петя волнуется, что где-то ошибся, поэтому просит вас перепроверить его вычисления.

Формат ввода

На первой строке вводятся 3 числа N, M, Q(1≤N×M≤3⋅105,1≤Q≤105) — количество строк, столбцов в таблице и количество ограничений.

В следующей строке вводятся через пробел M слов, состоящих из латинских маленьких букв — название соответствующей колонки, каждая строка по длине не превосходит L(1≤L≤10)

Далее вводятся N строк, в каждой через пробел M целых чисел aij(−109≤aij≤109) — элементы i-ой строки.

Потом вводятся Q строк — ограничения к запросу.

Каждая строка имеет вид ColumnNamekqkvalk(qk∈(<,>);−109≤valk≤109) — k-ое ограничение в формате, описанном в условии задачи.

Гарантируется, что ColumnNamek соответствует имени одной из колонок таблицы.

Формат вывода

Выведите единственное значение S — сумму всех чисел в строках, удовлетворяющих **всем**заданным ограничениям.

Если никакая строка не удовлетворяет всем запросам — выведите в ответ 0.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2 2 3  a b  1 1  2 2  a < 3  b > 1  b < 3 | 4 |

Примечания

Пояснение к примеру:

1. первый запрос: «a<3» оставляет все строки;
2. второй запрос: «b>1» оставляет только вторую строку;
3. третий запрос: «b<3» оставляет все строки.

В результате в ответе только вторая строка, сумма всех чисел в ней: 2+2=4

# 

# D. Межпланетная организация

|  | Все языки | Clang 16.0.0 C++20 | GNU GCC 12.2 C++20 |
| --- | --- | --- | --- |
| Ограничение времени | 5 секунд | 2 секунды | 2 секунды |
| Ограничение памяти | 512Mb | 512Mb | 512Mb |
| Ввод | стандартный ввод или input.txt | | |
| Вывод | стандартный вывод или output.txt | | |

Межпланетная организация имеет иерархическую древовидную структуру:

* Корнем иерархии является генеральный директор;
* У каждого сотрудника 0 или более непосредственных подчиненных;
* Каждый сотрудник, кроме генерального директора, является непосредственным подчиненным ровно одному сотруднику.

Каждый сотрудник, кроме генерального директора, говорит либо на языке A, либо на языке B. Директор говорит на двух языках для управления всей организацией.

Структура всей организации хранится в текстовом документе. Каждый сотрудник представлен уникальным идентификатором - целым числом от 0 до N включительно, где 0 - идентификатор генерального директора.

Каждый сотрудник представлен в документе ровно два раза. Между первым и вторым вхождением идентификатора сотрудника в аналогичном формате представлены все его подчиненные.

Если у сотрудника нет подчиненных, то два его идентификатора расположены один за другим.

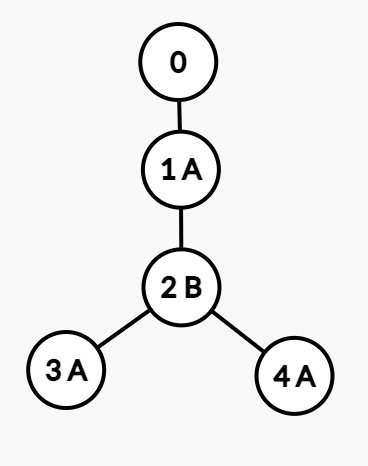
Например, если

* генеральный директор имеет в прямом подчинении сотрудника 1;
* сотрудник 1 имеет в прямом подчинении сотрудника 2;
* сотрудник 2 имеет в прямом подчинении сотрудников 3 и 4;

то документ будет представлен в виде строки:

0123344210

Если при этом сотрудники 1, 3, 4 говорят на языке A, а сотрудник 2 говорит на языке B, то вся организация выглядит так:



Назовем **языковым барьером**сотрудника X минимальное количество начальников между X и его начальником с таким же языком.

В нашем случае сотрудники 2, 3 и 4 - имеют языковой барьер 1, т.к. у каждого из них начальник говорит на неизвестном для них языке, а сразу следующий начальник говорит на том же языке, что они. В то же время сотрудник 1 имеет языковой барьер равный 0, поскольку его начальник - это директор, который знает два языка.

Вычислить языковой барьер для каждого сотрудника в компании.

## Формат ввода

В первой строке задано целое число N(1≤N≤106) — количество сотрудников (без генерального директора).

Во второй строке через пробел задано N символов Li(Li∈{A,B}) — язык i-го сотрудника.

В третьей строке через пробел задано 2⋅(N+1) целых чисел Pj(0≤Pj≤N) — иерархия организации в описанном в условии формате.

Гарантируется, что первый и последний элементы иерархии равны 0.

## Формат вывода

Выведите N целых чисел через пробел — языковой барьер каждого сотрудника от 1 до N включительно.

### **Пример 1**

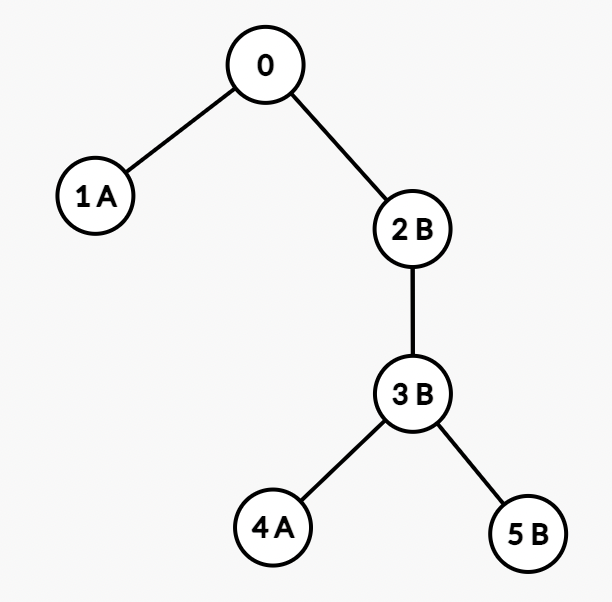
| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  A B B A B  0 1 1 2 3 4 4 5 5 3 2 0 | 0 0 0 2 0 |

### **Пример 2**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 4  A B A A  0 1 2 3 3 4 4 2 1 0 | 0 1 1 1 |

## Примечания

В первом примере все сотрудники кроме сотрудника 4 имеют нулевой языковой барьер. Сотрудник 4 имеет языковой барьер равный 2, т.к. ближайший начальник, говорящий с ним на одном языке - ген. директор, а между директором и данным сотрудником ровно 2 начальника.



# E. Близость

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Язык | Ограничение времени | Ограничение памяти | Ввод | Вывод |
| Все языки | 4 секунды | 256Mb | стандартный ввод или input.txt | стандартный вывод или output.txt |
| Golang 1.21.0 | 2 секунды | 256Mb |
| Clang 16.0.0 C++20 | 1 секунда | 256Mb |
| GNU GCC 12.2 C++20 | 1 секунда | 256Mb |

Определим **близость**двух целочисленных массивов как длину их наибольшего совпадающего префикса (см. примечание).

Примеры:

* Близость [1, 2, 1, 3] и [1, 2, 3, 2] равна 2 — префикс [1, 2] совпадает;
* Близость [1, 2, 3] и [3, 2, 1] равна 0.

Дано n целочисленных массивов a1,a2,…,an.

Необходимо вычислить сумму близостей массивов ai и aj для каждой пары 1≤i<j≤n.

## Формат ввода

Первая строка содержит одно целые число n(1≤n≤3⋅105)  — количество массивов.

Каждый массив задаётся двумя строками.

Первая строка описания массива содержит единственное целое число ki(1≤k≤3⋅105)  — размер i-го массива.

Вторая строка описания содержит ki целых чисел aij(1≤aij≤109) — элементы i-го массива.

Гарантируется, что ∑ni=1ki≤3⋅105.

## Формат вывода

Выведите единственное целое число  — суммарную попарную близость массивов.

### **Пример 1**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  2  1 2  2  1 3  3  1 2 3 | 4 |

### **Пример 2**

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  1  5  2  1 2  3  5 1 2 | 1 |

## Примечания

**Префиксом**длины p массива a назовем p первых элементов массива a.

Пример: у массива [1, 2, 1] есть 4 различных префикса:

1. Длины 0 — [] (пустой);
2. Длины 1 — [1];
3. Длины 2 — [1, 2];
4. Длины 3 — [1, 2, 1];

Рассмотрим близости массивов в **первом тестовом примере**:

* [1,2] и [1,3]: общий префикс [1] — близость равна 1;
* [1,2] и [1,2,3]: общий префикс [1,2] — близость равна 2;
* [1,3] и [1,2,3]: общий префикс [1] — близость равна 1;

Суммарно получается 1+2+1=4.

Рассмотрим близости массивов во **втором тестовом примере**:

* [5] и [1,2]: общий префикс [] — близость равна 0;
* [5] и [5,1,2]: общий префикс [5] — близость равна 1;
* [1,2] и [5,1,2]: общий префикс [] — близость равна 0;

Суммарно получается 0+1+0=1.