

Задача А. 4 элемента с суммой 0

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.6 секунд
Ограничение по памяти: 256 мегабайт

Дано четыре списка целых чисел A , B , C и D одинаковой длины n . Посчитайте количество четверок целых чисел (x, y, z, t) ($1 \leq x, y, z, t \leq n$) таких, что $A_x + B_y + C_z + D_t = 0$.

Формат входных данных

Первая строка содержит целое число n — количество чисел в списках ($1 \leq n \leq 2500$). Следующие n строк содержат четверки целых чисел A_i, B_i, C_i, D_i — i -е элементы каждого из четырех списков. Все элементы по абсолютному значению не превосходят 2^{28} .

Формат выходных данных

В единственной строке выведите ответ на данную задачу.

Пример

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 6 -45 22 42 -16 -41 -27 56 30 -36 53 -37 77 -36 30 -75 -46 26 -38 -10 62 -32 -54 -6 45 | 5 |

Замечание

Четверки элементов, образующие в примере сумму 0, следующие: $(-45; -27; 42; 30)$, $(26; 30; -10; -46)$, $(-32; 22; 56; -46)$, $(-32; 30; -75; 77)$, $(-32; -54; 56; 30)$

Задача В. Профили-двойники

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 second
Ограничение по памяти: 256 megabytes

Вам предложили работу в компании, разрабатывающей крупную социальную сеть. Ваше первое задание связано с поиском профилей, с большой вероятностью принадлежащих одному и тому же пользователю.

В социальной сети зарегистрировано n профилей, пронумерованных от 1 до n . Некоторые пары среди них являются друзьями (отношение «быть друзьями» взаимно, то есть если i является другом j , то и j является другом i). Будем говорить, что профили i и j ($i \neq j$) являются *двойниками*, если для любого профиля k ($k \neq i$, $k \neq j$), верно одно из двух утверждений: либо k дружит и с i , и с j , либо k не дружит ни с одним из них. При этом i и j могут как дружить между собой, так и не дружить.

Вам нужно посчитать количество различных неупорядоченных пар (i, j) , таких что профили i и j — двойники. Обратите внимание, что пары неупорядоченные, то есть пары (a, b) и (b, a) считается одинаковыми.

Формат входных данных

В первой строке записано два целых числа n и m ($1 \leq n \leq 10^6$, $0 \leq m \leq 10^6$), разделенных пробелом — количество профилей и количество пар друзей соответственно.

В следующих m строках записаны описания пар друзей в формате « v u », где v и u ($1 \leq v, u \leq n$, $v \neq u$) — номера профилей, являющихся друзьями. Гарантируется, что каждая неупорядоченная пара друзей встречается не более одного раза и никакой профиль не является другом самого себя.

Формат выходных данных

Выведите единственное целое число — количество неупорядоченных пар профилей, являющихся двойниками.

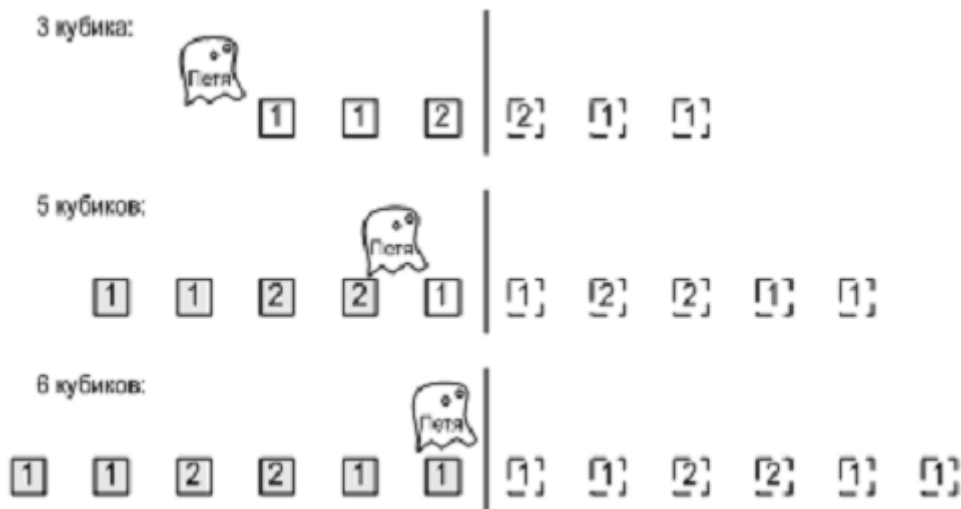
Примеры

| <code>stdin</code> | <code>stdout</code> |
|--------------------------|---------------------|
| 3 3 1 2 2 3 1 3 | 3 |
| 3 0 | 3 |
| 4 1 1 3 | 2 |

Задача С. Кубики

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать свое творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются. Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале. Помогите Пете! Выясните, сколько кубиков может быть у Пети. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.



Формат входных данных

Первая строка входного файла содержит число N ($1 \leq N \leq 10^6$) и количество различных цветов, в которые могут быть раскрашены кубики — M ($1 \leq M \leq 10^6$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Формат выходных данных

Выведите в выходной файл все такие K , что у Пети может быть K кубиков в подрядке возрастания

Пример

| стандартный ввод | стандартный вывод |
|--------------------|-------------------|
| 6 2 1 1 2 2 1 1 | 3 5 6 |

Задача D. Период строки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана непустая строка s . Нужно найти такое наибольшее число k и строку t , что s совпадает со строкой t , выписанной k раз подряд.

Формат входных данных

Одна строка длины N , ($1 \leq N \leq 10^6$), состоящая только из маленьких латинских букв.

Формат выходных данных

Одно число – наибольшее возможное k .

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| aaaaa | 5 |
| abcabcabc | 3 |
| abab | 2 |

Задача Е. Словарь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

Формат входных данных

В первой строке дан текст (не более 10^6 строчных латинских букв). Далее дано число M — количество слов в словаре.

В следующих M строках записаны слова (**каждое длиной не более 30 строчных латинских букв**). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более 10^5 .

Формат выходных данных

M строк вида Yes, если слово присутствует, и No иначе.

Примеры

| стандартный ввод | стандартный вывод |
|--------------------------------|-------------------|
| trololo | No |
| 3 | Yes |
| abacabadabacaba | Yes |
| olo | |
| trol | |
| abacabadabacabaabacabadabacaba | Yes |
| 1 | |
| abacabadabacabaabacabadabacaba | |

Задача F. Взлом хеширования

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

В этой задаче требуется найти коллизию при полиномиальном хешировании строк, состоящих из маленьких букв английского алфавита.

Полиномиальный хеш строки имеет два параметра: множитель p и модуль q . Для пустой строки ε значение хеш-функции $h(\varepsilon) = 0$, а для любой строки S и любого символа c хеш-функция рекуррентно определяется как $h(S + c) = (h(S) \cdot p + \text{code}(c)) \bmod q$. Здесь $\text{code}(c)$ — это ASCII-код символа c . Как известно, коды маленьких букв английского алфавита идут подряд: $\text{code}(\text{'a'}) = 97$, $\text{code}(\text{'b'}) = 98$, \dots , $\text{code}(\text{'z'}) = 122$. Можно выписать и нереккуррентную формулу: если строка $S = s_1 s_2 \dots s_n$, то $h(S) = (\text{code}(s_1) \cdot p^{n-1} + \text{code}(s_2) \cdot p^{n-2} + \dots + \text{code}(s_n) \cdot p^0) \bmod q$.

По заданным числам p и q найдите две различные непустые строки A и B такие, что $h(A) = h(B)$.

Формат входных данных

Первая строка ввода содержит два целых числа p и q , разделённых пробелом — параметры функции хеширования ($0 < p < q < 2 \cdot 10^{18}$).

Формат выходных данных

В первых двух строках выведите две различные непустые строки A и B , для которых $h(A) = h(B)$. Строки должны состоять исключительно из маленьких букв английского алфавита (ASCII-коды 97–122) и иметь длину от 1 до 100 000 символов. Заметим, что длины строк не обязательно должны совпадать. Если возможных ответов несколько, разрешается вывести любой из них.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 31 47 | aa bq |

Задача G. Неточное совпадение

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки с точностью до возможного несовпадения одного символа.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^5$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| aaaa | 4 |
| Caaabdaaaa | 1 2 6 7 |

Задача Н. Палиндромные заклинания

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Егор учится в школе чародейства и волшебства. Ни для кого не секрет, что там учатся колдовать. Помогите Егору с изучением заклинаний.

Любое заклинание — слово из маленьких латинских букв. Выучить большое заклинание тяжело, но его можно разбить на кусочки, каждый выучить отдельно, а потом собрать вместе. Склеить заклинание из кусочков можно только если симметричные кусочки совпадают. Такое разбиение называется палиндромным. Например, у `(d)(ec)(ode)` ровно 2 палиндромных разбиения: `(de)(co)(de)` и `(decode)`. Это также демонстрирует, что у любого заклинания есть тривиальное палиндромное разбиение размера 1. Простотой разбиения называется количество кусочков в нем. Найдите для Егора разбиение для заклинания s с максимальной простотой.

Формат входных данных

Первая строка содержит количество тестов t ($1 \leq t \leq 10$).

Следующие t строк содержат по одному заклинанию s . Каждая строка состоит из маленьких букв английского алфавита и имеет длину не более 10^6 .

Формат выходных данных

Для каждого теста выведите ответ.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 | 3 |
| bonobo | 5 |
| deleted | 7 |
| racecar | 1 |
| racecars | |

Замечание

- 1 строка: `(bo)(no)(bo)`
- 2 строка: `(d)(e)(let)(e)(d)`
- 3 строка: `(r)(a)(c)(e)(c)(a)(r)`
- 4 строка: `(racecars)`

Задача I. Массивы-палиндромы

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 512 мегабайт |

Кай работает в лаборатории изучения массивов, он экспериментирует с двумя массивами натуральных чисел: $A = [a_1, a_2, \dots, a_n]$ длины n и $B = [b_1, b_2, \dots, b_m]$ длины m .

Эксперимент, который проводит Кай, устроен следующим образом. У каждого из массивов отбрасывается произвольный, возможно пустой, префикс, а также произвольный, возможно пустой, суффикс, таким образом, чтобы оставшиеся части массивов имели равную длину. Обозначим получившиеся массивы как A' и B' , а их длину как k . Затем Кай суммирует поэлементно получившиеся массивы, итоговый массив Кай обозначает как $C = [c_1, c_2, \dots, c_k]$.

Пусть, например, $n = 5$, $A = [4, 3, 3, 2, 1]$, $m = 6$, $B = [4, 1, 5, 1, 3, 2]$, от массива A отбрасывается первый и последний элемент, от массива B три первых. После этого массивы имеют вид $A' = [3, 3, 2]$, $B' = [1, 3, 2]$, результат их поэлементного суммирования $C = [4, 6, 4]$.

Задача Кая заключается в том, чтобы получать такие C , которые являются *массивами-палиндромами*, то есть если числа на первой и последней позиции совпадают, числа на второй и предпоследней позиции совпадают, и так далее, для всех i числа на позициях i и $k - i + 1$ совпадают.

Помогите Каю понять, какой максимальный по длине массив-палиндром он может получить в результате эксперимента.

Формат входных данных

В первой строке ввода даны два целых числа n и m — количество элементов в первом и во втором массиве, соответственно ($1 \leq n, m \leq 100\,000$).

Во второй строке ввода даны n целых чисел a_i — массив A ($1 \leq a_i \leq 100$).

В третьей строке ввода даны m целых чисел b_j — массив B ($1 \leq b_j \leq 100$).

Формат выходных данных

Выведите единственное целое число — максимальное k , что Кай в результате эксперимента может получить массив-палиндром длины k .

Пример

| стандартный ввод | стандартный вывод |
|---------------------------------|-------------------|
| 5 6 4 3 3 2 1 4 1 5 1 3 2 | 3 |

Задача J. Родные просторы

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 1024 мегабайта |

Вы играете на смартфоне в игру «Родные просторы», в которой управляющий Остап помогает помещику восстановить отцовский дом. Игра происходит следующим образом.

Дана последовательность из n кристаллов, расположенных в один ряд слева направо. Каждый кристалл относится к одному из k видов, обозначенных первыми k английскими буквами. Таким образом, последовательность кристаллов записывается строкой английских букв.

За один ход игры можно удалить из последовательности один кристалл. Цель игрока — получить в результате применения разрешенных видов удалений лексикографически минимально возможную строку.

Разрешённые виды удаления кристаллов заданы таблицей A размера $k \times k$ из нулей и единиц. Если $A_{ij} = 1$, то разрешается удалить кристалл вида j , если непосредственно слева от него находится кристалл вида i . Данные действия можно выполнять в любом порядке.

Напомним, что строка x лексикографически меньше строки y , если выполнено одно из двух условий:

- существует такая позиция символа m , присутствующая в обеих строках, что до m -го символа строки совпадают, а m -й символ строки x меньше m -го символа y ,
- строка x является строгим префиксом y (то есть получается отбрасыванием одного или больше символов с конца строки y).

Формат входных данных

В первой строке даны два целых числа k и n ($1 \leq k \leq 26$, $1 \leq n \leq 500\,000$) — количество видов кристаллов и длина исходной последовательности кристаллов.

В следующих k строках задана таблица A , i -я строка содержит ровно k символов 0 или 1. Символ в i -й строке на j -й позиции равен A_{ij} .

В последней строке записаны n строчных английских букв, задающие исходную последовательность кристаллов. Гарантируется, что в строке встречаются только первые k букв английского алфавита, i -я по счёту буква английского алфавита обозначает i -й вид кристаллов.

Формат выходных данных

Выведите лексикографически минимальную строку, которую можно получить из исходной строки разрешёнными действиями.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------------------|-------------------|
| 3 7 010 001 100 abacaba | aac |
| 3 5 010 001 100 bcacb | bacb |

Задача К. С днём рождения!

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

В этой задаче требуется найти коллизию при полиномиальном хешировании строк, состоящих из маленьких букв английского алфавита.

Полиномиальный хеш строки имеет два параметра: множитель p и модуль q . Для пустой строки ε значение хеш-функции $h(\varepsilon) = 0$, а для любой строки S и любого символа c хеш-функция рекуррентно определяется как $h(S+c) = (h(S) \cdot p + \text{code}(c)) \bmod q$. Здесь $\text{code}(c)$ — это ASCII-код символа c . Как известно, коды маленьких букв английского алфавита идут подряд: $\text{code}(\text{'a'}) = 97$, $\text{code}(\text{'b'}) = 98$, ..., $\text{code}(\text{'z'}) = 122$. Можно выписать и нерекуррентную формулу: если строка $S = s_1s_2 \dots s_n$, то $h(S) = (\text{code}(s_1) \cdot p^{n-1} + \text{code}(s_2) \cdot p^{n-2} + \dots + \text{code}(s_n) \cdot p^0) \bmod q$.

По заданным числам p и q найдите две различные непустые строки A и B такие, что $h(A) = h(B)$.

Формат входных данных

Первая строка ввода содержит два целых числа p и q , разделённых пробелом — параметры функции хеширования ($0 < p < q \leq 2 \cdot 10^9 + 9$).

Формат выходных данных

В первых двух строках выведите две различные непустые строки A и B , для которых $h(A) = h(B)$. Строки должны состоять исключительно из маленьких букв английского алфавита (ASCII-коды 97–122) и иметь длину от 1 до 100 000 символов. Заметим, что длины строк не обязательно должны совпадать. Если возможных ответов несколько, разрешается вывести любой из них.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|----------------------|
| 31 47 | aa bq |
| 2 1000000007 | rp nx |
| 179 1000000009 | weeoutf hronndauw |