Задача A. Цепочка слов - 2

Имя входного файла:	chain2.in
Имя выходного файла:	chain2.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Цепочкой слов длины n назовем последовательность слов $w_1, w_2, \ldots, w_n$ такую, что для $1 \le i < n$ слово w_i является собственным префиксом слова w_{i+1}.

Напомним, что слово u длины k называется *собственным префиксом* слова v длины l, если $l > k$ и первые k букв слова v совпадают со словом u.

Задано множество слов $S = \{s_1, s_2, \ldots, s_m\}$. Найдите самую длинную цепочку слов, которую можно построить, используя (возможно, не все) слова этого множества.

Формат входного файла

Первая строка входного файла содержит целое число m ($1 \le m \le 250000$). Каждая из последующих m строк содержит по одному слову из множества S.

Все слова не пусты, имеют длину, не превосходящую 250000 символов, и состоят только из строчных букв латинского алфавита. Суммарная длина всех слов не превосходит 250000.

Формат выходного файла

В первой строке выходного файла выведите длину самой длинной цепочки слов, которую можно построить из (возможно, не всех) слов, заданных во входном файле.

Во второй строке выведите через пробел номера слов, образующих такую цепочку, в том порядке, в котором они в нее входят. Номера разделяйте пробелами. Слова нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Если решений несколько — выведите любое.

Пример

chain2.in	chain2.out
3 a ab abc	3 1 2 3
5 a ab bc bcd add	2 3 4

Задача B. Перевернутая система

Имя входного файла:	`inverse.in`
Имя выходного файла:	`inverse.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Слава учится в школе. Недавно на уроке математики он научился решать системы линейных уравнений. Напомним, что система n линейных уравнений с n неизвестными имеет следующий вид:

$$\begin{cases} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 \\ \ldots & + & \ldots & + & \ldots & + & \ldots & = & \ldots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \ldots & + & a_{nn}x_n & = & b_n \end{cases}$$

Научившись решать системы линейных уравнений, Слава решил не останавливаться на достигнутом и стал изучать системы уравнений других видов. Наиболее простыми ему показались системы следующего вида:

$$\begin{cases} (a_{11}x_1)^{-1} & + & (a_{12}x_2)^{-1} & + & \ldots & + & (a_{1n}x_n)^{-1} & = & b_1^{-1} \\ (a_{21}x_1)^{-1} & + & (a_{22}x_2)^{-1} & + & \ldots & + & (a_{2n}x_n)^{-1} & = & b_2^{-1} \\ \ldots & + & \ldots & + & \ldots & + & \ldots & = & \ldots \\ (a_{n1}x_1)^{-1} & + & (a_{n2}x_2)^{-1} & + & \ldots & + & (a_{nn}x_n)^{-1} & = & b_n^{-1} \end{cases}$$

где $x^{-1} = \frac{1}{x}$. Подумав немного, он понял, как решать такие системы, сводя их к системам линейных уравнений. Теперь Слава хочет проверить, правильно ли он их решает. Помогите ему найти правильный ответ!

Формат входного файла

В первой строке входного файла находится число n ($1 \le n \le 25$) — число переменных в системе уравнений.

В последующих n строках записано по $n+1$ целому числу. Первые n из этих чисел — это $a_{i1} \ldots a_{in}$, последнее число — это b_i, где i — это номер данной строки (первая строка входного файла не учитывается). Все числа a_{ij} и b_k не равны нулю и не превосходят 50 по абсолютной величине.

Гарантируется, что решение системы всегда существует, единственно, и ни один x_i не равен нулю.

Формат выходного файла

Выведите значения переменных $x_1 \ldots x_n$, по одному на каждой строке.

Обозначим правильные значения переменных за $c_1 \ldots c_n$. Ваш ответ будет засчитан, если для любого i ($1 \le i \le n$) будет верно хотя бы одно из следующих утверждений:

- $|x_i - c_i| \le 10^{-6}$
- $|x_i - c_i| \le 10^{-6} \cdot |c_i|$

Пример

inverse.in	inverse.out
3 1 2 3 5 2 1 3 4 2 3 1 3	24.4444444444 7.0967741935 3.7606837606
3 1 2 3 4 2 3 4 5 3 4 5 6	20.0000000000 -1.6666666666 0.6666666666
8 1 2 3 4 5 6 7 8 9 2 3 4 5 6 7 8 9 10 3 4 5 6 7 8 9 10 11 4 5 6 7 8 9 10 11 12 5 6 7 8 9 10 11 12 13 6 7 8 9 10 11 12 13 14 7 8 9 10 11 12 13 14 15 8 9 10 11 12 13 14 15 16	-12870.0000000000 178.7500000000 -10.2142857142 1.3928571428 -0.3714285714 0.1785714285 -0.1530612244 0.2500000000

Задача C. Кратчайший путь

Имя входного файла:	`path.in`
Имя выходного файла:	`path.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Одна планета далекой системы в соседней галактике имеет форму объединения двух сфер равного радиуса r километров. Вся поверхность планеты представляет собой пустыню, поэтому ее обитатели используют специальные сэндмобили для перемещения по поверхности. Сэндмобили двигаются по любой кривой на поверхности планеты, но не могут от нее удаляться. В частности, сэндмобили могут перемещаться внутри одной из сфер (планета полая внутри), оставаясь, однако, при этом на поверхности другой сферы.

Джон живет на этой планете. Однажды он решил посетить своего друга Джима. Помогите Джону найти кратчайшее расстояние от его дома до дома Джима.

Формат входного файла

Первая строка входного файла содержит два целых числа r и d ($0 < d < 2r \leq 1000$) — радиусы сфер, из которых состоит планета, и расстояние между их центрами. Пусть координатная система введена таким образом, что центры расположены в точках $(0, 0, 0)$ и $(0, 0, d)$.

Вторая строка содержит три вещественных числа x_a, y_a и z_a — координаты дома Джона. Третья строка также содержит три вещественных числа x_b, y_b и z_b — координаты дома Джима. Каждая точка находится на расстоянии r от центра одной из сфер и на расстоянии строго больше r от центра другой сферы. Все равенства выполнены с точностью до 10^{-9}.

Формат выходного файла

Выведите одно целое число — длину кратчайшего пути между домами Джона и Джима вдоль поверхности планеты. Ваш ответ должен отличаться от правильного не более чем на 10^{-5}.

Пример

path.in	path.out
5 8 0 0 -5 0 0 13	24.9809154479650885

Задача D. Черное и белое

Имя входного файла:	segments.in
Имя выходного файла:	segments.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Вдоль прямой расположено несколько чередующихся черных и белых отрезков целой длины. Каждую секунду выбирается самый длинный отрезок (если есть несколько самых длинных отрезков, выбирается самый левый из них) и изменяется, используя следующие правила:

- Если выбранный отрезок белый, то он разбивается на две части, правая часть перекрашивается в черный цвет, а левая остается белой.
- Если выбранный отрезок черный, то он разбивается на две части, левая часть перекрашивается в белый цвет, а правая остается черный.

После этого если рядом оказывается два отрезка одного цвета, то они объединяются в один.

При разбиении отрезок четной длины делится на две равные части. Если же разбивается отрезок нечетной длины, то длины получившихся частей — максимально близкие целые числа, причем часть, которая сохраняет цвет исходного отрезка оказывается длиннее.

Процесс продолжается до тех пор, пока не выполнится одно из следующих условий: все отрезки будут иметь длину 1, либо некоторая конфигурация отрезков повторяется дважды.

Вам задана начальная конфигурация отрезков и число t. Выясните, в течение какого времени будет продолжаться процесс, либо выведите, что процесс не завершится за t секунд.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и t ($1 \le n \le 100\,000$, $1 \le t \le 100\,000$) — количество отрезков в начальной конфигурации и количество шагов, которые требуется просимулировать.

Вторая строка содержит слово “`white`”, если самый левый отрезок в начальной конфигурации белый, или слово “`black`”, если самый левый отрезок в начальной конфигурации черный.

Третья строка содержит n целых чисел — длины отрезков в начальной конфигурации, слева направо, в порядке, в котором они следует вдоль прямой. Длина каждого отрезка не превышает 10^9.

Формат выходного файла

Выведите количество секунд, которые будет продолжаться процесс, либо t, если процесс не завершится за t секунд.

Пример

segments.in	segments.out
2 4 white 5 4	3
2 10 white 1000000000 1	10