Задача A. Двоичные последовательности

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

Необходимо найти количество двоичных последовательностей (последовательностей, в которых каждый символ является либо нулем, либо единицей) длины n, в которых заданная строка s встречается как подстрока ровно один раз.

Формат входного файла

Первая строка строка входного файла содержит число n ($1 \le n \le 10^4$). Вторая строка входного файла содержит непустую строку s из нулей единиц. Ее длина не превышает 100 символов.

Формат выходного файла

В выходной файл выведите остаток от деления искомого количества последовательностей на $10^9 + 7$.

Примеры

binary.in	binary.out
4 01	10
8 0	8

Задача B. Произведение цифр

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

Найдите k-ое в порядке возрастания число, произведение цифр десятичной записи которого является степенью двойки.

Формат входного файла

Первая строка входного файла содержит целое число k ($1 \le k \le 10^{18}$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

product.in	product.out
1	1
10	22

Задача C. В чем истина?

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

Известный логический парадокс, связанный с утверждением, содержащем информацию о самом себе, заключен в предложении «Это утверждение ложно». Попытка выяснить, является ли это утверждение истинным или ложным, приводит к противоречию. Обобщим этот пример.

Рассмотрим последовательность утверждений, каждое из которых имеет одну из трех форм: «*Следующее утверждение истинно.*», «*Следующее утверждение ложно.*» или «*В последовательности ровно n истинных утверждений.*». Считается, что первое утверждение следует после последнего.

Требуется определить, можно ли таким образом назначить каждому утверждению истинно оно ложно, чтобы все истинные утверждения были действительно истинны, а все ложные утверждения — ложны.

Формат входного файла

Первая строка входного файла содержит число m — количество утверждений ($1 \le m \le 100\,000$). Каждая из следующих m строк содержит информацию о соответствующем утверждении:

- "`+`" — означает, что соответствующее утверждение «*Следующее утверждение истинно.*»;
- "`-`" — означает, что соответствующее утверждение «*Следующее утверждение ложно.*»;
- "`$` n" — означает, что соответствующее утверждение «*В последовательности ровно n истинных утверждений.*» ($0 \le n \le 100\,000$).

Формат выходного файла

Если система противоречива, выведите на первой строка выходного файла слово «`inconsistent`». В противном случае выведите «`consistent`». В этом случае на второй строке выведите m символов, i-й из них должен быть «`t`», если i-е утверждение истинно, и «`f`», если оно ложно. Если имеется несколько допустимых назначений, можно вывести любое.

Пример

truth.in	truth.out
3 + + $ 3	consistent ttt
3 + - $ 3	consistent ttf
1 -	inconsistent

Задача D. Сад камней

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

Сад камней обычно представляет собой небольшую огороженную территорию, на которой находится несколько камней. Посетители сада камней садятся на траву, расслабляются и созерцают камни. Некоторые сады камней отличаются тем, что лишь из немногих секретных точек можно наблюдать одновременно все камни.

Юный японец Нитеринети Йолимипийади недавно нашел новый сад камней, и теперь хочет узнать, есть ли в нем точка, из которой можно целиком наблюдать все камни. Сад, найденный Нитеринети, довольно особенный — все камни в нем представляют собой высокие круглые колонны. Сад имеет прямоугольную форму, его размер — $x \times y$ метров. Мальчик нарисовал план сада на листе бумаги и начал искать секретное место. Помогите ему выяснить, есть ли в саду место, из которого каждый камень полностью виден — а именно, любой луч, проведенный из заданной точки, пересекает внутренность не более чем одного камня (но может касаться нескольких камней). Точка не должна быть внутри камня (но может быть на его границе).

Формат входного файла

Первая строка входного файла содержит три целых числа: n, x и y — количество камней в саду и размеры сада ($1 \le n \le 10$, $4 \le x, y \le 10^4$).

Введем систему координат таким образом, чтобы углы сада располагались в точках $(0, 0)$, $(x, 0)$, $(0, y)$ и (x, y). Следующие n строк содержат по три целых числа — координаты центров камней и их радиусы. Все камни находятся внутри сада, никакие два камня не пересекаются (хотя они могут касаться друг друга).

Формат выходного файла

Если внутри сада есть точка, из которой полностью видны все камни, выведите ее координаты на первой строке выходного файла. В противном случае выведите “`No Zen`”. Выводите как можно больше знаков после десятичной точки. Проверяющая программа будет использовать точность 10^{-6} при сравнении вещественных чисел.

Пример

zen.in	zen.out
4 10 10 2 2 2 8 8 2 2 8 2 8 2 2	5.0 5.0
5 10 10 2 2 2 8 8 2 2 8 2 8 2 2 5 5 1	No Zen