Задача A. Магазин

Имя входного файла:	`shop.in`
Имя выходного файла:	`shop.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы работаете менеджером самого среднего звена в магазине. Ваш босс поручил вам провести рекламную акцию для привлечения покупателей в магазин. Рекламная акция заключается в том, что при покупке одного товара, покупатель в подарок получает другой товар.

Вам дан товар, к которому будет предоставлен подарок. Но этот подарок должен иметь стоимость не менее a процентов и не более, чем b процентов, от стоимости товара. Для выгоды магазина стоимость подарка должна быть при этом минимально возможной.

Вам необходимо выбрать этот подарок для заданного товара.

Формат входного файла

В первой строке входного файла задано название товара, к которому вы должны выбрать подарок. Во второй строке задано натуральное число t ($1 \le t \le 10^9$) — стоимость товара. В третьей строке заданы два натуральных числа a и b ($1 \le a \le b \le 100$). В четвёртой строке задано натуральное число n ($1 \le n \le 1000$) — количество товаров-кандидатов на подарок.

Далее следуют описания этих товаров. Каждый товар описывается в двух строках. Первая строка содержит название кандидата. Во второй строке задано натуральное число x_i ($1 \le x_i \le 10^9$) — его стоимость.

Гарантируется, что стоимости всех товаров-кандидатов на подарок различны. Длины названий товаров не превышают 20 символов и названия могут содержать символы с кодами от 32 до 127, то есть строчные и заглавные латинские буквы, знаки препинания, арифметические знаки, цифры и пробелы.

Формат выходного файла

В единственной строке выходного файла выведите название товара, который будет подарком. Гарантируется, что такой товар существует.

Примеры

shop.in	shop.out
Coca-Cola 2l+2l. 120 15 25 3 Coca-Cola 0,5l. 20 Dirol 14 Ice-cream 40	Coca-Cola 0,5l.

Примечание

Решения, работающие при стоимости товаров до 10^3, будут оцениваться из 40 баллов.

Задача B. Спутник

Имя входного файла:	`satellite.in`
Имя выходного файла:	`satellite.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Компания «РосПрог» занимается написанием программного обеспечения для спутников. Спутники летают быстро, поэтому и программы на нём должны работать быстро (иначе он может не успеть, например, рассчитать и подкорректировать свою траекторию). У разработчиков есть несколько различных реализаций функции расчёта траектории, и они хотят выбрать самую быструю из них.

Для этого они взяли k реализаций, запустили каждую по n раз на тестовом стенде и измерили, сколько времени эти реализации каждый раз работали. После этого для каждой пары реализаций a и b было посчитано *доминирование* a над b. *Доминированием* реализации a над реализацией b называется количество пар запусков реализаций a и b таких, что запуск реализации a отработал строго быстрее запуска реализации b.

После этого была посчитана *производительность* каждой реализации. *Производительность* реализации a определяется как сумма доминирований a над всеми реализациями, кроме a. Из посчитанных данных должен быть составлен отчёт для начальства, но в последний день перед сдачей данные были потеряны. Помогите разработчикам всё-таки сдать отчёт начальству и восстановите значения всех производительностей.

Формат входного файла

В первой строке задано два числа n и k ($1 \le n, k \le 1000$) — количество запусков и количество различных реализаций, соответственно. Далее, в k строках задано по n целых чисел $a_{i,j}$ ($1 \le a_{i,j} \le 10^9$) — время работы j-го запуска i-й реализации.

Все $a_{i,j}$ различны.

Формат выходного файла

В первой и единственной строке выведите k чисел. i-е число должно равняться производительности i-й реализации.

Примеры

satellite.in	satellite.out
3 3 1 4 7 2 5 8 3 6 9	12 9 6
3 1 2 3 9	0

Примечание

Решения, работающие при $n, k \le 100$, будут оцениваться из 40 баллов.

Решения, работающие при $n, k \le 300$, будут оцениваться из 60 баллов.

Задача C. Древний календарь

Имя входного файла:	`calendar.in`
Имя выходного файла:	`calendar.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Недавно Вася прочитал статью о том, что берляндские ученые обнаружили древний календарь исчезнувшей цивилизации туземцев апрелья, который, как и многие другие древние календари, предсказывал не столь далекий конец света.

Календарь туземцы создали, опираясь на свои священные числа. Нулевым священным числом была единица, первым — тоже единица, а каждое следующее священное число F_i вычислялось как сумма двух предыдущих: $F_i = F_{i-1} + F_{i-2}$. Каждое натуральное число в календаре представлялось в виде суммы некоторых чисел $F_i, i > 0$ таким образом, что если в сумме присутсвовало некоторое F_j, то ни F_{j-1}, ни F_{j+1} в разложении не присутствовало. В такой системе счисления, например, число 16 имело вид 100100, так как $16 = 1 \cdot 13 + 0 \cdot 8 + 0 \cdot 5 + 1 \cdot 3 + 0 \cdot 2 + 0 \cdot 1$. Затем все числа, записанные таким образом, последовательно перенесли на календарь, начиная с единицы. Если обозначить запись числа i в описанной системе счисления как $f\{i\}$, то календарь представляет собой строку $S = f\{1\}f\{2\}f\{3\}\ldots$, состоящую из записей $f\{i\}$ всех натуральных чисел.

А вас Вася просит определить, сколько групп из двух подряд идущих нулей находится в промежутке с первого символа календаря по N-й символ.

Формат входного файла

Во входном файле содержится единственное число N ($1 \le N \le 10^{18}$) — номер последнего символа, который интересен Васе.

Формат выходного файла

В выходной файл выведите единственное число — количество вхождений групп из двух подряд идущих нулей среди первых N символов календаря.

Примеры

calendar.in	calendar.out
20	4

Примечание

Первые 20 символов календаря — 11010010110001001101.

Решения, работающие при $N \le 10^6$, будут оцениваться из 40 баллов.

Задача D. Меч

Имя входного файла:	`sword.in`
Имя выходного файла:	`sword.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Когда-то у маленького Васи был игрушечный меч и он им часто хвастался перед друзьями. Но сегодня произошло неисправимое: меч куда-то потерялся! Он в панике пошел искать потерянную игрушку. Не прошло и часа, как Вася отыскал пропажу. Теперь он хочет вернуть меч на его законное место. Но игрушка была очень тяжелой, и именно поэтому он ею не мог играть, а только хвастался. Так как Вася еще маленький, вам предстоит помочь ему доставить пропажу на свое место.

Будем считать, что меч и заготовленное под него место — это отрезки на плоскости равной и ненулевой длины. За один ход Вася может переместить меч только так, чтобы один из его концов остался на своем месте, так как Вася не может полностью поднять игрушку.

Ваша задача определить, за какое минимальное количество ходов Вася сможет доставить свой меч на его законное место и как именно ему нужно действовать для достижения этой цели.

Формат входного файла

В первой строке входного файла заданы четыре целых числа: x_1, y_1, x_2 и y_2 — координаты позиции, где обычно находился меч.

Во второй строке в таком же формате заданы координаты концов меча в текущем положении.

Все числа во входном файле по модулю не превосходят 10^9. Длина меча не равна нулю.

Формат выходного файла

В выходной файл выведите последовательность положений меча в процессе его возвращения на место. Каждая строка соответствует очередному его положению. Координаты концов меча можно выводить в любом порядке, но сначала обе координаты одного конца, потом другого. Длина последовательности должна быть минимально возможной. Первая строка соответствует начальному положению меча, последняя — месту, где он должен лежать.

Все сравнения на равенство будут проводиться с абсолютной или относительной погрешностью 10^{-6}. Гарантируется, что длина последовательности не превышает 1000.

Примеры

sword.in	sword.out
`1 1 2 2` `0 0 1 1`	`0 0 1 1` `1 1 2 2`

`0 0 2 0`
`1 0 1 2`
	`1 0 1 2`
	`1 0 0.5 -1.9364916731`
	`0.5 -1.9364916731 0 0`
	`0 0 2 0`

Примечание

Решения, работающие для тестов, когда все координаты $y = 0$, будут оцениваться из 60 баллов.