Задача 5. Кондиционер

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

В офисе, где работает программист Петр, установили кондиционер нового типа. Этот кондиционер отличается особой простотой в управлении. У кондиционера есть всего лишь два управляемых параметра: желаемая температура и режим работы.

Кондиционер может работать в следующих четырех режимах:

- «freeze» — охлаждение. В этом режиме кондиционер может только уменьшать температуру. Если температура в комнате и так не больше желаемой, то он выключается.
- «heat» — нагрев. В этом режиме кондиционер может только увеличивать температуру. Если температура в комнате и так не меньше желаемой, то он выключается.
- «auto» — автоматический режим. В этом режиме кондиционер может как увеличивать, так и уменьшать температуру в комнате до желаемой.
- «fan» — вентиляция. В этом режиме кондиционер осуществляет только вентиляцию воздуха и не изменяет температуру в комнате.

Кондиционер достаточно мощный, поэтому при настройке на правильный режим работы он за час доводит температуру в комнате до желаемой.

Требуется написать программу, которая по заданной температуре в комнате t_{room}, установленным на кондиционере желаемой температуре t_{cond} и режиму работы определяет температуру, которая установится в комнате через час.

Формат входного файла

Первая строка входного файла содержит два целых числа t_{room}, и t_{cond}, разделенных ровно одним пробелом ($-50 \le t_{room} \le 50$, $-50 \le t_{cond} \le 50$).

Вторая строка содержит одно слово, записанное строчными буквами латинского алфавита — режим работы кондиционера.

Формат выходного файла

Выходной файл должен содержать одно целое число — температуру, которая установится в комнате через час.

Примеры входных и выходных файлов

cond.in	cond.out
10 20 heat	20
10 20 freeze	10

Пояснения к примерам

В первом примере кондиционер находится в режиме нагрева. Через час он нагреет комнату до желаемой температуры в 20 градусов.

Во втором примере кондиционер находится в режиме охлаждения. Поскольку температура в комнате ниже, чем желаемая, кондиционер самостоятельно выключается и температура в комнате не поменяется.

Задача 6. Праздничный ужин

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

Рядом с офисом компании, в которой работает программист Джон, открылось новое кафе. Директор компании решил провести там новогодний ужин.

Меню праздничного новогоднего ужина в кафе состоит из k типов блюд. Для каждого типа блюда есть несколько вариантов на выбор. Всего есть a_1 вариантов для первого типа блюда, a_2 вариантов для второго типа блюда, и так далее, a_k вариантов для k-го типа блюда. Всего, таким образом, предлагается $a_1 \times a_2 \times \ldots \times a_k$ различных заказов праздничного ужина.

Всего на ужине будут присутствовать m сотрудников компании. Каждый сотрудник должен заказать ровно один вариант блюда каждого типа на выбор. Таким образом, ужин каждого сотрудника будет состоять из k блюд. Для того чтобы ужин каждого сотрудника компании был уникален, администратор кафе придумал следующую схему. Сотрудники делают заказ ужина из меню один за другим. Каждый сотрудник выбирает k блюд, по одному варианту каждого типа. После выбора заказа из меню, сотрудник указывает один их типов блюд, и выбранный этим сотрудником вариант блюда этого типа больше не предлагается тем сотрудникам, которые делают заказ после него.

Каждый сотрудник компании запомнил, сколько возможных заказов ужина ему было предложено. Выяснилось, что директору, который выбирал первым, было предложено на выбор $n_1 = a_1 \times a_2 \times \ldots \times a_k$ заказов. Тому, кто выбирал вторым, досталось лишь $n_2 < n_1$ заказов, поскольку один из вариантов одного из типов блюд уже не был доступен, и так далее. Джону, который выбирал последним, был предложен выбор лишь из n_m заказов. Джон заинтересовался, а какое количество вариантов каждого типа блюд было на выбор у директора компании.

Требуется написать программу, которая по заданным числам k, m и n_1, n_2, …, n_m выяснит, какое количество вариантов каждого типа блюд изначально предлагалось на выбор.

Формат входного файла

Первая строка входного файла содержит два целых числа k и m, разделенных ровно одним пробелом ($1 \le k \le 20$, $2 \le m \le 100$). Вторая строка содержит m чисел: n_1, n_2, …, n_m (для всех i от 1 до m выполняется неравенство $1 \le n_i \le 10^9$).

Формат выходного файла

Выходной файл должен содержать k чисел: a_1, a_2, …, a_k. Если возможных вариантов решения поставленной задачи несколько, требуется вывести любой. Соседние числа должны быть разделены ровно одним пробелом. Гарантируется, что хотя бы одно решение существует.

Пример входного и выходного файлов

dinner.in	dinner.out
3 3 12 8 4	3 2 2

Пояснения к примеру

События в примере могли развиваться, например, следующим образом.

Исходно количество заказов ужина было равно 3×2×2=12. Директор, выбрав свой заказ, указал блюдо первого типа, поэтому второму сотруднику осталось лишь два варианта блюда первого типа. Количество заказов для него сократилось до 2×2×2 = 8. Он также указал на свое блюдо первого типа, и Джон уже мог выбирать лишь из 1×2×2 = 4 заказов ужина.

Задача 7. Космический кегельбан

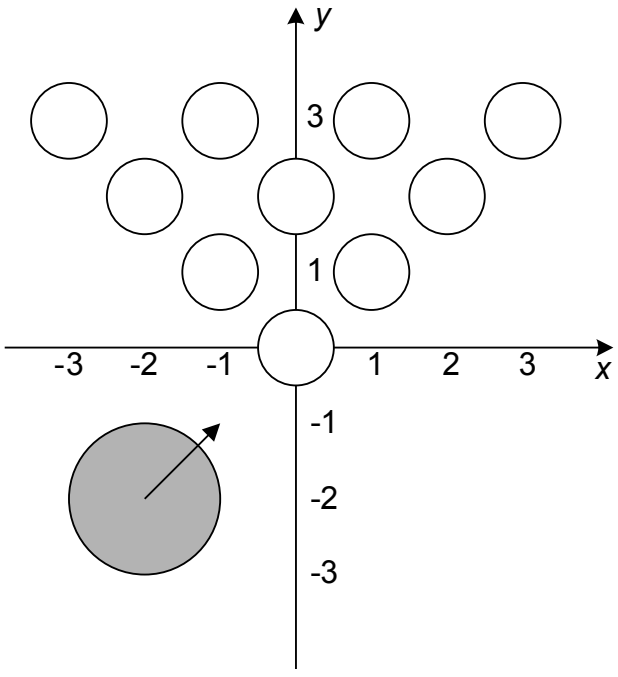
Имя входного файла:	`spacepin.in`
Имя выходного файла:	`spacepin.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На планете Плюк открылся новый космический кегельбан. Поле для кегельбана представляет собой бесконечную плоскость, на которой расставлены кегли.

Каждая кегля представляет собой высокий цилиндр с основанием в виде круга радиусом r метров. Все кегли одинаковые. Кегли расставлены по следующим правилам. Кегли образуют n рядов, в первом ряду стоит одна кегля, во втором — две, и так далее. В последнем n-м ряду стоит n кеглей. Введем на плоскости систему координат таким образом, чтобы единица измерения была равна одному километру. Центр единственной кегли в первом ряду находится в точке (0, 0). Центры кеглей во втором ряду находятся в точках (–1, 1) и (1, 1). Таким образом, центры кеглей в i-м ряду находятся в точках с координатами (–(i – 1), i – 1), (–(i – 3), i – 1), …, (i – 1, i – 1).

Игра происходит следующим образом. Используется шар с радиусом q метров. Игрок выбирает начальное положение центра шара (x_c, y_c) и вектор направления движения шара (v_x, v_y). После этого шар помещается в начальную точку и двигается, не останавливаясь, в направлении вектора (v_x, v_y). Считается, что шар сбил кеглю, если в процессе движения шара имеет место ситуация, когда у шара и кегли есть общая точка. Сбитые кегли не меняют направления движения шара и не сбивают соседние кегли при падении.

На рисунке приведен пример расположения кеглей для $r = 500$, $n = 4$ и шара для $q = 1000$, $x_c = –2$, $y_c = –2$, $v_x = 1$, $v_y = 1$.



Требуется написать программу, которая по заданным радиусу кегли r, количеству рядов кеглей n, радиусу шара q, его начальному положению (x_c, y_c) и вектору направления движения (v_x, v_y) определяет количество кеглей, сбитых шаром.

Формат входного файла

Первая строка входного файла содержит два целых числа: r и n, разделенных ровно одним пробелом ($1 \le r \le 700$, $1 \le n \le 200\,000$).

Вторая строка входного файла содержит целое число q ($1 \le q \le 10^9$).

Третья строка входного файла содержит два целых числа x_c и y_c, разделенных ровно одним пробелом ($-10^6 \le x_c \le 10^6$, $-10^6 \le y_c$, $1000 \times y_c < -(r + q)$).

Четвертая строка входного файла содержит два целых числа v_x и v_y, разделенных ровно одним пробелом ($-10^6 \le v_x \le 10^6$, $0 < v_y \le 10^6$).

Формат выходного файла

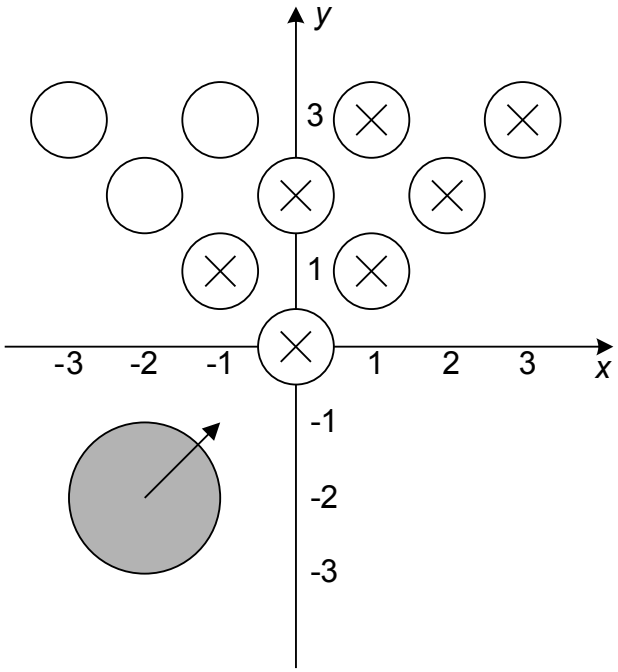
Выходной файл должен содержать одно целое число — количество сбитых кеглей.

Пример входного и выходного файлов

spacepin.in	spacepin.out
500 4 1000 -2 -2 1 1	7

Пояснения к примеру

Рисунок ниже показывает, какие кегли будут сбиты (такие кегли обозначены «x»).



Система оценивания

Правильные решения для тестов, в которых $1 \le n \le 1000$ и $v_x = 0$, будут оцениваться из 20 баллов.

Правильные решения для тестов, в которых $1 \le n \le 1000$ и $v_x \ne 0$, будут оцениваться еще из 20 баллов.

Правильные решения для тестов, в которых $1000 < n \le 200\,000$ и $v_x = 0$, будут оцениваться еще из 20 баллов.

Чтобы получить оставшиеся 40 баллов, решение должно правильно работать также для тестов, в которых $1000 < n \le 200\,000$ и $v_x \ne 0$.

Задача 8. «Abracadabra»

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

Строка s называется *супрефиксом* для строки t, если t начинается с s и заканчивается на s. Например, «abra» является супрефиксом для строки «abracadabra». В частности, сама строка t является своим супрефиксом. Супрефиксы играют важную роль в различных алгоритмах на строках.

В этой задаче требуется решить обратную задачу о поиске супрефикса, которая заключается в следующем. Задан словарь, содержащий n слов $t_1, t_2, \ldots, t_n$ и набор из m строк-образцов $s_1, s_2, \ldots, s_m$. Необходимо для каждой строки-образца из заданного набора найти количество слов в словаре, для которых эта строка-образец является супрефиксом.

Требуется написать программу, которая по заданному числу n, n словам словаря $t_1, t_2, \ldots, t_n$, заданному числу m и m строкам-образцам $s_1, s_2, \ldots, s_m$ вычислит для каждой строки-образца количество слов из словаря, для которых эта строка-образец является супрефиксом.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 200\,000$).

Последующие n строк содержат слова $t_1, t_2, \ldots, t_n$, по одному слову в каждой строке. Каждое слово состоит из строчных букв латинского алфавита. Длина каждого слова не превышает 50. Суммарная длина всех слов не превышает 10^6. Словарь не содержит пустых слов.

Затем следует строка, содержащая целое число m ($1 \le m \le 200\,000$).

Последующие m строк содержат строки-образцы $s_1, s_2, \ldots, s_m$, по одной на каждой строке. Каждая строка-образец состоит из строчных букв латинского алфавита: Длина каждой строки-образца не превышает 50. Суммарная длина всех строк-образцов не превышает 10^6. Никакая строка-образец не является пустой строкой.

Формат выходного файла

Выходной файл должен содержать m чисел, по одному на строке.

Для каждой строки-образца в порядке, в котором они заданы во входном файле, следует вывести количество слов словаря, для которых она является супрефиксом.

Пример входного и выходного файлов

sufpref.in	sufpref.out
4 abacaba abracadabra aa abra 3 a abra abac	4 2 0

Система оценивания

Правильные решения для тестов, в которых $1 \le n \le 100$, $1 \le m \le 100$, будут оцениваться из 30 баллов.