Задача A. Лара Крофт и клад

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Лара Крофт нашла глиняную табличку, на которой было написана следующая инструкция:

- Пройдите m метров на север от места, где обнаружена табличка.
- После этого пройдите m метров на восток.
- После этого пройдите $\sqrt{2} \cdot m$ метров на юго-запад.
- Здесь и находится клад.

По заданному m вычислите, на каком расстоянии от места обнаружения глиняной таблички находится клад.

Формат входных данных

Входные данные содержат одно целое число m ($1 \leqslant m \leqslant 1000$) — расстояние с таблички.

Формат выходных данных

Выведите одно число — расстояние от глиняной таблички до клада в метрах, округлённое до ближайшего целого.

Пример

стандартный ввод	стандартный вывод
2	0

Задача B. История ICPC

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Хотя история финалов чемпионатов по программированию ICPC отсчитывается с 1977 года, современный вид соревнования приобрели только в 1992. Дело в том, что до финала 1991 года включительно в команду на соревнованиях входило не трое, а четверо студентов.

Представьте, что вы — тренер студенческих команд в начале 1991-1992 учебного года. В конце прошлого сезона в тренировках участвовало K полных команд по старым правилам; из участников этих команд L закончили обучение в университете, а с нового набора ваши тренировки посещает M студентов. Какое наименьшее количество студентов вам надо дополнительно привлечь к тренировкам, чтобы к локальному чемпионату, который пройдёт уже по новым правилам, ни один из тренирующихся студентов не остался без полноценной команды?

Формат входных данных

Входной файл содержит три целых числа — K, L и M ($1 \leqslant K \leqslant 100$, $0 \leqslant L \leqslant 4 \cdot K$, $1 \leqslant M \leqslant 100$).

Формат выходных данных

Выведите одно число — наименьшее количество студентов, которых надо дополнительно привлечь к тренировкам, чтобы к первому контесту сезона 1991-1992 ни один из тренирующихся студентов не остался без команды.

Примеры

стандартный ввод	стандартный вывод
3 2 1	1
20 1 8	0

Задача C. Петя, Маша и верёвочки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

На столе лежали две одинаковые верёвочки целой положительной длины.

Петя разрезал одну из верёвочек на N частей целой положительной длины, так что на столе стало $N+1$ верёвочек. Затем в комнату зашла Маша и взяла одну из лежащих на столе верёвочек. По длинам оставшихся на столе N верёвочек определите, какую **наименьшую** длину может иметь верёвочка, взятая Машей.

Формат входных данных

Первая строка входных данных содержит одно целое число N — количество верёвочек, оставшихся на столе ($2 \leqslant N \leqslant 1000$). Каждая из последующих N строк содержит по одному целому числу l_i — длине очередной верёвочки ($1 \leqslant l_i \leqslant 1000$).

Формат выходных данных

Выведите одно целое число — наименьшую длину, которую может иметь верёвочка, взятая Машей.

Примеры

стандартный ввод	стандартный вывод
4 1 5 2 1	1
4 5 12 4 3	24

Задача D. Многозначность

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Сколько существует таких целых положительных чисел, которые в B_1-ичной системе счисления являются D_1-значными числами, а в B_2-ичной — D_2-значными?

Формат входных данных

Первая строка входных данных содержит два целых числа B_1 и D_1 ($2 \leqslant B_1 \leqslant 100$, $1 \leqslant D_1 \leqslant 20$). Вторая строка содержит два целых числа B_2 и D_2 ($2 \leqslant B_2 \leqslant 100$, $1 \leqslant D_2 \leqslant 20$).

Гарантируется, что наибольшее D_1-значное число в B_1-ичной системе счисления и наибольшее D_2-значное число в B_2-ичной системе счисления не превосходят 10^{18}.

Формат выходных данных

Выведите одно число — количество целых положительных чисел, являющихся D_1-значными в B_1-ичной системе счисления и D_2-значными в B_2-ичной системе счисления.

Примеры

стандартный ввод	стандартный вывод
3 1 5 1	2
10 2 2 4	6

Замечание

В первом примере из условия требуется найти количество чисел, являющихся однозначными в системах счисления с основаниями 3 и 5. Таких чисел 2: 1 и 2 (0 не является положительным числом). Во втором примере требуется найти количество двузначных чисел в десятичной системе счисления, которые являются четырёхзначными в двоичной. Таких чисел 6: от 10_{10} (1010_2) до 15_{10} (1111_2) включительно.

Задача E. Неактуальный баннер

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Как известно, с этого года командный чемпионат мира по программированию называется ICPC; название же "ACM" более не актуально и использоваться не должно.

Однако дизайнер, делавший баннер к очередному сезону ICPC, скопировал старый дизайн и изобразил в углу баннера таблицу $M \times N$, заполненную заглавными латинскими буквами '`A`', '`C`' и '`M`'. Заказчик решил снизить вознаграждение дизайнера на сумму, равную количеству способов прочитать слово "`ACM`" на баннере.

Считается, что в таблице можно прочитать слово "`ACM`" если в некоторой клетке записана буква "`A`", в одной из соседних с ней по стороне — буква "`C`", а в одной из соседних уже с ней по стороне — буква "`M`". Способы считаются различными, если хотя бы одна из клеток, задействованных при прочтении слова этими способами, различается.

Ваша задача — определить, на какую сумму заказчик снизит вознаграждение дизайнеру.

Формат входных данных

Первая строка входа содержит два целых числа M и N — размерность таблицы ($1 \leqslant M \leqslant 1000$, $1 \leqslant N \leqslant 1000$). Каждая из последующих M строк содержит по N заглавных латинских букв '`A`', '`C`' или '`M`'.

Формат выходных данных

Выведите одно число — сумму, на которую заказчик снизит вознаграждение дизайнеру.

Пример

стандартный ввод	стандартный вывод
2 4 ACMA AACA	4

Замечание

В примере к задаче слово можно прочитать четырьмя способами. Если координаты верхнего левого поля обозначим как $(1, 1)$, то возможные способы:

1. $(1, 1) - (2, 1) - (3, 1)$;
2. $(2, 2) - (2, 1) - (3, 1)$;
3. $(2, 2) - (3, 2) - (3, 1)$;
4. $(4, 2) - (3, 2) - (3, 1)$.

Задача F. Гиперсферы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Гиперсферой радиуса R с центром O в N-мерном пространстве называется множество всех точек, находящихся **ровно** на расстоянии R от точки O. Напоминаем, что расстояние между точкой a с координатами x_1, x_2, …, x_N и точкой b с координатами y_1, y_2, …, y_N в N-мерном пространстве равно

$$\sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$$

Даны две N-мерные гиперсферы с целыми радиусами, все координаты центров которых также целые. Выясните, сколько у них общих точек.

Формат входных данных

Первая строка входных данных содержит одно целое число N — размерность пространства ($2 \leqslant N \leqslant 100$). Вторая строка содержит параметры первой гиперсферы — радиус R и координаты центра x_1, x_2, …, x_N ($1 \leqslant R \leqslant 10^5$, $-10^5 \leqslant x_i \leqslant 10^5$). Третья строка содержит параметры второй гиперсферы в аналогичном формате.

Формат выходных данных

Выведите одно целое число — количество общих точек двух гиперсфер или -1, если общих точек бесконечно много.

Примеры

стандартный ввод	стандартный вывод
3 2 0 0 0 1 3 0 0	1
3 2 0 0 0 1 1 1 1	-1
3 2 0 0 0 2 0 1 8	0

Задача G. Баскетбольная статистика

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В Национальной Баскетбольной Ассоциации для каждого игрока считается статистика, в том числе набранных очков, результативных передач, подборов и перехватов.

Байтландская Баскетбольная Ассоциация также решила ввести такую статистику. Правила баскетбола в Байтландии несколько отличаются от существующих в NBA; важная для статистики часть будет описана далее.

После каждого матча аналитики сдают протокол, который в упрощённом виде можно представить в виде следующего хронологически упорядоченного списка:

- Владение — игрок X из команды T ($T = 0$ или $T = 1$) владел мячом в соответствующий момент времени. Обозначается как “`BALL X(T)`”

- Фол — на игроке X из команды T были нарушены правила, что привело к назначению Pen штрафных бросков по кольцу соперника, пробиваемых этим игроком. Обозначается как “`FOUL ON X(T) Pen`”, где Pen — количество бросков.

- Штрафной бросок — игрок, **на котором нарушили правила**, пробил штрафной бросок по кольцу соперника. После фола следующие Pen событий — это штрафные броски. Обозначаются как “`FREE Res`”, где Res — 0, если игрок не попал в кольцо, и 1, если попал. Больше ни в каких ситуациях штрафные броски встретиться не могут.

- Бросок с игры — владеющий мячом игрок бросил по кольцу соперника с расстояния L сантиметров. Обозначается как “`RING L Res`”, где Res — 0, если игрок не попал, и 1, если игрок попал.

Верны следующие ограничения:

- Владение завершается либо сменой игрока, владеющего мячом, либо фолом, либо броском по кольцу, то есть две подряд записи “`BALL`”, относящиеся к одному и тому же игроку, невозможны.

- После фола с N штрафными бросками обязательно следуют N бросков подряд (серия штрафных).

- После попадания в кольцо с игры и после попадания с последнего штрафного в серии мяч обязательно переходит игроку другой команды (то есть следующее событие — “`BALL`” со сменой номера команды).

- После неудачного броска или неудачного последнего штрафного в серии следующее событие — обязательно владение (то есть “`BALL`”).

Статистика определяется следующим образом:

- Очки: за попадание со штрафного начисляется 1 очко, за попадание с расстояния, меньшего или равного 675 сантиметров — 2 очка, за попадание с расстояния, большего 675 сантиметров — 3 очка.

- Передачи: если бросок был забит с игры и владение, предшествующее завершившемуся броском владению, было у игрока той же команды, что и бросавший игрок, то соответствующему игроку засчитывается результативная передача. Иначе говоря, для начисления передачи игроку $P1$ в протоколе должны подряд идти записи “`BALL P1(T)`”, “`BALL P2(T)`” и “`RING L 1`”.

- Перехваты: если владение переходит к другой команде без броска или фола, игроку команды, получившей мяч, засчитывается перехват. Иначе говоря, для начисления перехвата игроком $P1$ в протоколе должны подряд идти записи “`BALL P0(1-T)`” и “`BALL P1(T)`”.

- Подборы: первое владение после неудачного броска (с игры или **последнего** в серии штрафных) засчитывается как подбор. Иначе говоря, для начисления подбора игроком $P1$ в протоколе должны подряд идти записи “`RING L 0`” или “`FREE 0`” и “`BALL P1(T)`”, причём по какому кольцу был бросок, значения не имеет.

Ваша задача — по протоколу матча определить статистику каждого игрока.

Формат входных данных

Первая строка входных данных содержит целое число N_1 — количество задействованных в процессе игры игроков в команде 0 ($5 \leqslant N_1 \leqslant 10$). В следующих N_1 строках задаются имена игроков команды 0. Каждое имя представляет собой непустую строку из заглавных латинских букв длиной не более 20 символов. Далее идёт целое число N_2 ($5 \leqslant N_2 \leqslant 10$) — количество задействованных в процессе игры игроков в команде 1. Последующие N_2 строк задают имена игроков команды 1 в аналогичном формате. Гарантируется, что внутри одной команды все имена игроков различны.

Далее следует целое число Q ($1 \leqslant Q \leqslant 10^4$) — количество записей в протоколе. Каждая из последующих Q строк содержит одну запись в формате, описанном в условии.

- `BALL` $pname(team)$ — игрок с именем $pname$ из команды $team$ владел мячом ($0 \leqslant team \leqslant 1$).

- `FOUL ON` $pname(team)$ Pen — на игроке с именем $pname$ из команды $team$ были нарушены правила, игрок должен пробить Pen штрафных бросков ($0 \leqslant team \leqslant 1$, $1 \leqslant Pen \leqslant 3$).

- `RING` L Res — игрок, владевший мячом, бросил по кольцу соперника с L сантиметров с результатом Res ($1 \leqslant L \leqslant 1500$, $0 \leqslant Res \leqslant 1$). Если $Res = 1$, бросок был успешным, если $Res = 0$ — неудачным.

- `FREE` Res — игрок, на котором были нарушены правила, пробил штрафной бросок по кольцу соперника с результатом Res ($Res = 1$ — бросок был успешным, $Res = 0$ — неудачным).

Гарантируется, что протокол начинается с владения одной из команд, что описанные в условиях правила соблюдаются и что назначенное количество штрафных всегда будет пробито (то есть матч не может закончиться посередине серии штрафных бросков).

Формат выходных данных

Выведите список игроков в следующем формате: $TeamID$ $Name$ S_1 S_2 S_3 S_4, где $TeamID$ — номер команды (0 или 1), $Name$ — имя игрока, S_1 — количество очков, S_2 — количество передач, S_3 — количество перехватов и S_4 — количество подборов. Список должен быть отсортирован сначала по номеру команды, а затем лексикографически по имени игрока.

Пример

стандартный ввод	стандартный вывод
5 CURRY DURANT GREEN IGOUDALA PACHULIA 5 LEBRON IRVING KORVER SMITH LOVE 19 BALL IRVING(1) BALL LEBRON(1) RING 200 1 BALL CURRY(0) RING 1000 1 BALL SMITH(1) BALL GREEN(0) BALL DURANT(0) RING 400 0 BALL IGOUDALA(0) BALL CURRY(0) FOUL ON DURANT(0) 2 FREE 1 FREE 0 BALL LOVE(1) BALL IRVING(1) BALL LEBRON(1) BALL KORVER(1) RING 676 1	0 CURRY 3 0 0 0 0 DURANT 1 0 0 0 0 GREEN 0 0 1 0 0 IGOUDALA 0 0 0 1 0 PACHULIA 0 0 0 0 1 IRVING 0 1 0 0 1 KORVER 3 0 0 0 1 LEBRON 2 1 0 0 1 LOVE 0 0 0 1 1 SMITH 0 0 0 0

Задача H. Игра с матрицей

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

Алиса учит Боба работать с матрицами. Недавно она рассказала Бобу, что такое определитель.

Напомним, что *перестановкой* называется набор чисел от 1 до n, расположенных в некотором порядке.

Инверсией в перестановке π называется такая пара чисел i, j, что $i < j$ и $\pi_i > \pi_j$

Определителем квадратной матрицы A размера n называется $\sum\limits_{\pi \in P_n} (-1)^{N(\pi)} A_{1,\pi_1} \cdot A_{2,\pi_2} \cdot \ldots \cdot A_{n,\pi_n}$, где $N(\pi)$ — количество инверсий в перестановке π, P_n — множество всех перестановок размера n.

Боб не до конца усвоил урок, и поэтому Алиса разрешила ему поиграть с квадратной матрицей размера n. Боб может за одно действие прибавить произвольное число к произвольному элементу матрицы. В ответ Алиса называет ему определитель полученной матрицы.

Так как Алиса не умеет перемножать в уме большие числа, она делает все операции по модулю $10^9 + 7$.

Бобу стало скучно просто прибавлять числа к элементам матрицы, поэтому он решил насолить Алисе. Он хочет добиться того, чтобы определитель матрицы стал нулем. Ваша задача — помочь ему в этом.

Протокол взаимодействия

В начале вашего взаимодействия с программой жюри вам следует считать целых два числа — n и a ($1 \leqslant n \leqslant 20$; $1 \leqslant a < 10^9 + 7$) — размер матрицы и ее определитель.

После этого начинается взаимодействие. Каждый ваш запрос описывается тремя числами x, y, d ($1 \leqslant x, y \leqslant n$; $1 \leqslant d < 10^9 + 7$) — номер строки и столбца элемента, который вы хотите изменить, а также число, которое вы хотите к нему прибавить.

На каждый ваш запрос программа жюри в отдельную строку записывает единственное число a ($0 \leqslant a < 10^9 + 7$) — определитель получившейся матрицы.

После того, как вы получили ответ $a = 0$, ваша программа до Как только ваша программа получила ответ $a = 0$, она должна завершить работу.

Обратите внимание, что у Алисы не так много времени, так что она не будет выполнять более 30 запросов.

Пример

стандартный ввод	стандартный вывод
3 2 5 21 221 220 0	1 2 3 2 3 4 3 1 10 2 2 1000000006 1 2 1000000003

Замечание

Разберём первый тест. В нем компьютер загадал матрицу, которая выглядит следующим образом:

```
1 1 0
0 1 1
1 0 1
```

После первых трёх ходов матрица выглядит так:

```
1 4 0
0 1 5
```

```
11 0 1
```

После этого мы к центральному элементу прибавляем число 10^9+6, но поскольку все операции у нас производятся по модулю 10^9+7, то центральный элемент становится равен нулю. Аналогичная ситуация происходит на последнем шаге, и итоговая матрица выглядит так:

```
1 0 0
0 0 5
11 0 1
```

Несложно убедиться, что её определитель равен нулю.

Задача I. Прямые и треугольники

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Даны целые положительные числа n и k. Необходимо провести n прямых, таких, что:

- никакие две прямые не являются параллельными или совпадающими;
- никакие три прямые не пересекаются в одной точке;
- среди частей, на которые прямые разобьют плоскость, ровно k будут треугольниками.

Формат входных данных

В единственной строке через пробел записаны два целых положительных числа n и k, $3 \leqslant n \leqslant 50$, $n - 2 \leqslant k \leqslant n$.

Формат выходных данных

Выведите n прямых, по одной на каждой строке. Каждая прямая задается четырьмя целыми числами x_1, y_1, x_2, y_2, не превосходящими по абсолютной величине 10 000 и записанными через пробел; здесь (x_1, y_1) и (x_2, y_2) - две различные точки, лежащие на задаваемой прямой.

Если соответствующие прямые подобрать невозможно, выведите в единственной строке число -123456789. Гарантируется, что если удовлетворяющие условию n прямых существуют, то существуют и удовлетворяющие условию n прямых, которые могут быть заданы вышеописанным способом.

Примеры

стандартный ввод	стандартный вывод
3 1	0 0 3 5 3 5 6 -1 6 -1 0 0
3 2	-123456789

Задача J. Камни, компьютер и вечность

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Снежная Королева постигает азы теории игр при помощи развивающей компьютерной программы. В ней человек играет против компьютера. Правила игры просты — изначально есть три кучки камней, в каждой лежит целое положительное количество камней. На каждом шагу компьютер делает следующее:

1. Выбирает кучку, в которой количество камней наибольшее.
2. Проверяет, верно ли, что количество камней в этой кучке не меньше, чем суммарно в двух других.
3. Если это так, то он убирает из выбранной кучки количество камней, равное суммарному количеству камней в других кучках, и переходит к шагу 1. Иначе завершает свой ход.

Снежная Королева на каждом своем ходу удваивает количество камней в какой-либо кучке.

Первым ходит компьютер.

Цель компьютера — добиться того, чтобы хотя бы одна кучка опустела. Снежной Королеве необходимо перевести игру в вечность, чтобы игра продолжалась бесконечно и размер любой кучки в любой момент времени был ненулевым.

Снежная Королева застряла на первом же уровне. Известно, что изначально в первой кучке может быть от 1 до a камней, во второй — от 1 до b, а в третьей — от 1 до c камней. Получается, что всего в игре $a \cdot b \cdot c$ уровней. Вам надо определить, в скольких из них выиграет Снежная Королева.

Две позиции считаются различными, если существует кучка, количество камней в которой в обеих позициях различно.

Формат входных данных

В единственной строке записаны три целых числа a, b, c ($1 \leqslant a, b, c \leqslant 2000$) — ограничения на размер первой, второй и третей кучки соответственно.

Формат выходных данных

Выведите единственное целое число — количество позиций, в которых побеждает Снежная Королева.

Пример

стандартный ввод	стандартный вывод
2 4 3	5

Замечание

В первом тесте позицией, в которой победит Снежная Королева, является, например, позиция 2 3 3. На первом ходу компьютер ничего сделать не может, поскольку 3 < 2 + 3, значит ход сразу переходит к Снежной Королеве. Снежная Королева может удвоить первую кучку, получив позицию 4 3 3. Компьютер по прежнему ничего не может сделать, поэтому Снежная Королева снова может удвоить первую кучку, получив позицию 8 3 3. На этот раз компьютер может сделать ход, поскольку $3 + 3 \leqslant 8$. После хода компьютера получается позиция 2 3 3, с которой начиналась игра, а значит, Снежная Королева может бесконечно ходить по этому циклу.

Позиция 1 3 3 также является выигрышной, поскольку компьютер пропускает свой первый ход, а Снежная Королева удваивает первую кучку и сводит позицию к 2 3 3.

Проигрышной для Снежной Королевы позицией является, например, позиция 1 2 1, в которой компьютер первым же ходом обнуляет вторую кучку и побеждает. Обратите внимание — 1 2 1 и 1 1 2 — это разные позиции!

Задача K. Игра с массивом

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

На последних соревнованиях по программированию Чипу подарили массив из n элементов. Чип долго с ним играл, и в итоге выделил в нем подпоследовательность. Он показал ее Гаечке, которая поняла, что подпоследовательность является k-й в лексикографическом порядке среди всех подпоследовательностей массива. К сожалению, после этого пришел Дейл и отобрал у Чипа его подпоследовательность.

Гаечка не любит, когда Чип расстраивается, и поэтому хочет вернуть ему его подпоследовательность. К сожалению, сейчас она слишком занята, и просит вас помочь ей.

По данным n, k и массиву из n элементов найдите в нем k-ю в лексикографическом порядке подпоследовательность.

Напомним, что подпоследовательность последовательности x_n — это последовательность x_{n_t}, где n_t — возрастающая последовательность.

Последовательность x лексикографически меньше последовательности y, если существует такое i, что $x_j = y_j$ для любого $j < i$, а $x_i < y_i$, либо если x является «началом» y.

Формат входных данных

В первой строке входных данных записаны два целых числа — n и k ($1 \leqslant n \leqslant 5 \cdot 10^5$; $1 \leqslant k \leqslant 10^{18}$) — количество элементов массива и номер искомой подпоследовательности соответственно.

Во второй строке записаны n целых чисел, по модулю не превосходящих 10^9 — элементы массива.

Формат выходных данных

Если в массиве меньше чем k подпоследовательностей и ответа не существует, выведите единственное число -1.

Иначе в единственной строке выведите ответ — искомую подпоследовательность.

Примеры

стандартный ввод	стандартный вывод
5 6 1 2 3 4 5	1 2 3 5
5 1 1 2 3 4 5	1
5 3 1 1 1 1 1	1 1 1
5 10 1 1 1 1 1	-1

Задача L. Граф. Просто граф...

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Дан неориентированный связный граф, состоящий из n вершин, пронумерованных от 1 до n, и m ребер, а также q запросов. В каждом из запросов, вам требуется по заданным различным вершинам u и v, найти количество различных простых путей, соединяющих u и v. Напомним, что путь называется простым, если этот путь проходит по каждой из вершин графа не более чем по одному разу.

Формат входных данных

В первой строке записаны целые неотрицательные числа n и m ($1 \leqslant n \leqslant 200\,000$, $0 \leqslant m \leqslant n + 4$ - количество вершин и рёбер графа.

Далее в следующих m строках записано по паре целых чисел u, v, $1 \leqslant u, v \leqslant n$, $u \neq v$.

В следующей строке записано целое число q, $1 \leqslant q \leqslant 2\,000\,000$ - количество запросов. В следующих строках записано по паре целых чисел u, v, $1 \leqslant u, v \leqslant n$, $u \neq v$, описывающих запрос.

Гарантируется, что в графе не существует петель и кратных ребер.

Формат выходных данных

Для каждого запроса выведите в отдельной строке количество различных путей между вершинами u и v

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 3 3 1 3 1 2 3 1 2 3	2 2 2
4 4 1 2 2 3 3 1 1 4 6 1 2 1 3 1 4 2 3 2 4 3 4	2 2 1 2 2 2