

Задача А. Союзы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В Триаметистовом королевстве было n городов и m дорог, соединявших некоторые из них друг с другом. Однажды в результате экспериментов придворного мага Д произошло катастрофическое расщепление: во вселенной вместо одного Триаметистового королевства появилось множество его копий, или *отражений*. Более того, в каждом из них каждая дорога стала заколдованной либо способом α , либо способом ω (стоит отметить, что каждое из возможных сочетаний заколдованностей дорог появилось ровно в одном из отражений).

Свойства заколдованностей α и ω таковы, что города a , b и c образуют торговый союз тогда и только тогда, когда есть дороги между a и b , между b и c и между c и a , заколдованные одним и тем же способом.

Формат входных данных

В первой строке входных данных записаны два целых числа n и m — количество городов и дорог Триаметистового королевства. В следующих m строках записаны пары чисел, задающие города, соединённые соответствующими дорогами. $3 \leq n \leq 10\,000$, $1 \leq m \leq 100\,000$. Никакие два города не соединены более чем одной дорогой, никакая дорога не соединяет город сам с собой.

Формат выходных данных

Выведите как можно точнее единственное вещественное число — среднее количество союзов, которое образовалось во всех отражениях Триаметистового королевства.

Пример

стандартный ввод	стандартный вывод
3 3	0.25
1 2	
2 3	
3 1	

Задача В. Японский компьютер

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как известно, для обороны границ японские инженеры разрабатывают огромных боевых человекоподобных роботов. Каждый такой робот управляется японским компьютером. Понятно, что для повышения эффективности робота программа в компьютере должна быть как можно более оптимальной, чтобы компьютер мог выполнять как можно больше программ за как можно меньшее время.

На данный момент японским программистам задали следующую задачу (её смысл секретен, поэтому здесь его описывать нельзя): изначально в памяти компьютера находится единственное число x . Требуется получить в его памяти следующие числа: a_1x, a_2x, \dots, a_nx . При этом компьютер может выполнять следующие операции:

1. Сложение двух чисел
2. Вычитание двух чисел
3. Побитовый сдвиг влево (сдвиг на k бит эквивалентен умножению на 2^k)

Все полученные промежуточные значения сохраняются в памяти, так что ими можно пользоваться при вычислении других значений.

При вычислениях никогда не должно получаться значение большее, чем $42x$. Гарантируется, что при выполнении этого ограничения, в компьютере не происходит переполнений. Также, компьютер не может работать с отрицательными числами, так что вычитать большее число из меньшего также запрещено.

Порядок, в котором в памяти будут появляться числа a_1x, a_2x, \dots, a_nx , не имеет значения.

Формат входных данных

В первой строке находится число n — количество требуемых значений ($1 \leq n \leq 41$). Во второй строке находится n чисел a_i ($2 \leq a_i \leq 42$). Все a_i различны. Само число x вам не дано, так что ваша последовательность операций должна быть верной для любого x .

Формат выходных данных

В первой строке выведите единственное число — минимальное количество требуемых операций. Далее выведите требуемые операции в следующем формате:

1. Сдвиг влево ax на k бит: “ $a<<k$ ”
2. Сложение ax и bx : “ $a+b$ ”
3. Вычитание ax из bx : “ $b-a$ ”

Запись операций не должна содержать пробелов.

Примеры

стандартный ввод	стандартный вывод
3 3 5 18	5 1<<1 1<<4 1+2 2+3 2+16
1 29	4 1<<1 1<<5 1+2 32-3
4 12 19 41 42	8 1<<1 1<<4 1+2 3<<2 3+16 19<<1 3+38 1+41

Задача С. Произведение графов

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пусть дан ориентированный ациклический граф. Стандартная игра на графе заключается в следующем: изначально на одной из вершин графа (называемой

начальной позицией) стоит фишка. Двое игроков по очереди двигают её по рёбрам. Проигрывает тот, кто не может сделать ход.

В теории игр часто рассматриваются более сложные игры. Например, прямое произведение двух игр на графах. Прямое произведение игр — это следующая игра: изначально на каждом графе в начальной позиции стоит по фишке. За ход игрок двигает обе фишки по рёбрам (каждую фишку двигает в собственном графе). Проигрывает тот, кто не может сделать ход. То есть тот, кто не может сделать ход хотя бы в одной игре.

Ваша задача — опеределить, кто выиграет при правильной игре.

Формат входных данных

На первой строке будут даны числа N_1 и M_1 — количество вершин и рёбер в первом графе ($1 \leq N_1, M_1 \leq 100\,000$). На следующих M_1 строках содержится по два числа x и y ($1 \leq x, y \leq N_1$).

В следующих $M_2 + 1$ строках задан второй граф в том же формате.

Заканчивается входной файл списком пар начальных вершин, для которых нужно решить задачу. На первой строке задано число T ($1 \leq T \leq 100\,000$) — количество пар начальных вершин. В следующих T строках указаны пары вершин v_1 и v_2 ($1 \leq v_1 \leq N_1, 1 \leq v_2 \leq N_2$).

Учтите, что в графах могут быть кратные рёбра.

Формат выходных данных

На каждую из T пар начальных вершин выведите строку “**first**”, если при правильной игре выиграет первый, и “**second**”, если второй.

Пример

стандартный ввод	стандартный вывод
3 2	first second
1 2	
2 3	
2 1	
1 2	
2	
1 1	
3 2	

Задача D. Путь

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На плоскости есть две точки и непересекающиеся стенки-квадратики. Нужно найти кратчайший путь из одной точки в другую.

Формат входных данных

В первых двух строках заданы координаты точек начала и конца пути. В третьей строке дано число N ($1 \leq N \leq 1000$) — количество квадратилов. Следующие N строк содержат описания квадратилов. Каждый квадратик задаётся координатами противоположных углов, а стороны его параллельны осям координат. Все координаты даны в формате $x\ y$, где x и y — целые числа, не превосходящие по модулю 10^9 .

Пояснения:

1. По стороне квадратила можно ходить. Если два квадратила соприкасаются стороной, между ними можно пройти.
2. Каждый квадратик — не просто прямоугольник, а именно квадрат.
3. Точки начала и конца не лежат внутри никакого квадратила, поэтому ответ всегда существует.
4. Квадратики не имеют общих внутренних точек.
5. Все квадратики имеют ненулевую площадь.
6. Точки начала и конца не совпадают.

Формат выходных данных

В первой строке нужно вывести длину найденного пути. Далее надо вывести сам путь в виде ломаной. Во второй строке выведите k — количество вершин в ломаной (число k не должно превосходить 10 000). В следующих k строках выведите координаты вершин ломаной в порядке прохода по ним (координаты должны быть целыми и по модулю не превосходить 10^9). Первая вершина должна совпадать с началом пути, последняя — с концом пути. Длина ломаной должна отличаться от выведенной и от кратчайшей не более чем на 10^{-6} . Если кратчайших ломаных несколько, разрешается выводить любую из них.

Пример

стандартный ввод	стандартный вывод
0 0	5.82842712474
5 0	4
2	0 0
1 -1 4 2	1 -1
1 -1 3 -3	4 -1
	5 0