Задача A. Transportation Problem

Имя входного файла:	`transport.in`
Имя выходного файла:	`transport.out`
Ограничение по времени:	6 seconds
Ограничение по памяти:	256 Mebibytes

Flat Oil company has recently won a tender for building an oil transportation system between a oil production points and b main oil consumers of Flatland. As a result a set E of m oil pipelines was built. All pipelines are designed to be unidirectional, so oil can only be transported in one (predefined) direction along the pipeline. In order to simplify the process of planning oil transportation (and since funding through the tender was quite impressive) each pipeline has daily capacity c equal to the daily production of oil in Flatland. The system is planned in such a way that it is possible to transport oil from any point to any other point (no matter, production or consumption).

However, when time came for planning transportation, it turned out that due to the extremely large size of the network, the problem is still quite difficult. So they needed external help. Help the company to create the oil transportation plan.

For each oil production point u its daily production p_u is known. For each oil consumption point u its daily consumption q_u is known. The sum of all p_u is equal to the sum of all q_u and is equal to the capacity of each pipeline c. Any production or consumption point can also be a transit point for oil transportation.

Oil transportation plan assigns each pipeline uv a *flow* $f(uv)$, so that the following conditions are satisfied:

- for each pipeline uv: $0 \le f(uv) \le c$;
- if u is a production point, $\sum\limits_{uv \in E} f(uv) - \sum\limits_{vu \in E} f(vu) = p_u$;
- if u is a consumption point, $\sum\limits_{vu \in E} f(vu) - \sum\limits_{uv \in E} f(uv) = q_u$.

Given the plan of the network, find the oil transportation plan for it.

Формат входного файла

The first line of the input file contains three integer numbers: a, b and m ($1 \le a, b \le 50\,000$, $1 \le m \le 100\,000$). The second line contains a integer numbers: p_i ($1 \le p_i \le 10^4$). The third line contains b integer numbers: q_i ($1 \le q_i \le 10^4$). The sum of all p_i is equal to the sum of all q_i. The following m lines describe pipelines, each line contains two integer numbers: s and t, ranging from $-a$ to b except 0 — the number of the point the pipeline starts at and the number of the point the pipeline ends at. The production points are numbered from -1 to $-a$ in the decreasing order in order their production values are given in the second line. The consumption points are numbered from 1 to b according to the order their consumption values are given in the third line.

Формат выходного файла

If a transportation plan exists, output "`YES`" at the first line of the output file. The second line must contain m integer numbers: the flow in the pipelines in order they are described in the input file.

If the transportation plan doesn't exist, output "`NO`".

Пример

transport.in	transport.out
2 2 6 5 3 4 4 -1 1 -2 1 -2 2 1 2 2 -1 2 -2	YES 5 0 3 1 0 0

Задача B. Разрезание прямоугольника-2

Имя входного файла:	`cutrect2.in`
Имя выходного файла:	`cutrect2.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 Mebibytes

После разрезания прямоугольника из предыдущей задачи Васю попросили разрезать ещё один прямоугольник для Института добавления и отрезания.

Этот прямоугольник размера $m \times n$ также вырезан из листа клетчатой бумаги по линиям сетки. Но теперь некоторые клетки прямоугольника (как минимум одна) объявлены важными. Вася должен разрезать этот прямоугольник прямой линией в заданном направлении так, чтобы отношение общей площади важных клеток в получающихся частях стало равно заданному.

Можете ли вы снова ему помочь?

Формат входного файла

Входной файл содержит не более чем две тысячи тестовых наборов. Каждый тестовый набор начинается со строки, содержащей три числа m, n и R, где m и n целые числа от 1 до 100, а R — это отношение площади левой части к площади правой части. «Левая» и «правая» части определены в соответствии с направлением ненулевого вектора направления разреза. Вектор направления разреза задаётся во второй строке двумя целыми числами dx и dy. Оба этих числа не превосходят 1000 по абсолютному значению. Остальные m строк тестового набора содержат прямоугольную карту: n символов в каждой строке, где '`*`' обозначает важную клетку, а '`.`' обозначает неважную клетку. Входной файл завершается строкой из тTJx нулей.

Гарантируется, что R — это вещественное число между 0.001 и 1000, заданное с максимально возможной точностью. Общее число клеток во всех тестовых наборах в одном входном файле не превосходит $2 \cdot 10^6$.

Формат выходного файла

Для каждого тестового набора выведите координаты одной из точек на линии разреза с максимально возможной точностью. Для вывода придерживайтесь формата вывода, показанного в примере ниже. Левый нижний угол прямоугольника имеет координаты $(0, 0)$.

Допускается абсолютная ошибка в определении отношения важных площадей не более 10^{-6} в предположении, что ваш ответ выведен с максимальной возможной точностью.

Примеры

cutrect2.in	cutrect2.out
1 1 1.0 -1 1 * 1 1 7.0 -1 1 * 1 1 7.0 1 -1 * 2 3 9.0 2 -3 *.* .** 0 0 0	Cut rectangle 1 at (0.5, 0.5) Cut rectangle 2 at (1, 0.5) Cut rectangle 3 at (0, 0.5) Cut rectangle 4 at (-0.011237390046160782, 2)

Задача C. Вложение графа

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

Институт красивых строк (ИКС) получил просьбу о помощи от Матрично-графового университета (МГУ). Темой исследования является вложение произвольных неориентированных графов в бесконечные бинарные деревья. Более подробно задача описана ниже.

Пусть $\mathcal{G} = (V, E)$ — граф, вершины которого занумерованы от 1 до n включительно. Пусть $\mathcal{T}$ — бесконечное корневое бинарное дерево, узлы которого помечены в соответствии со следующими правилами:

- Корень помечен ε (пустая строка).
- Для каждого $t \in \mathcal{T}$ его левый сын помечен $N(t) \cdot$ '`l`', а правый сын помечен $N(t) \cdot$ '`r`', где $N(t)$ — это метка вершины t, а операция '$\cdot$' — это конкатенация.

Таким образом, сыновья корневого узла помечены '`l`' и '`r`', их четыре сына помечены '`ll`', '`lr`', '`rl`' и '`rr`', и т. д.

Вложение графа $\mathcal{G}_1 = (V_1, E_1)$ в другой граф $\mathcal{G}_2 = (V_2, E_2)$ — это отображение $f : V_1 \to V_2$ такое, что выполняются следующие условия:

- Различные вершины $\mathcal{G}_1$ отображаются в различные вершины $\mathcal{G}_2$;
- Для каждого ребра $e_1 \in E_1$, соединяющего u и v, существует ребро $e_2 \in E_2$, соединяющее $f(u)$ и $f(v)$.

Другими словами, мы инъективно отображаем V_1 в подмножество V_2 так, что E_1 соответствует некоторому подмножеству E_2.

Будем обозначать вложение f графа $\mathcal{G}$ в дерево $\mathcal{T}$ как $(f(v_1), f(v_2), \ldots f(v_n))$ — список строк, которые соответствуют узлам в которые отображены вершины $1, 2, \ldots, n$.

Например, если $V = \{1, 2, 3\}$, $E = \{(1, 2), (1, 3)\}$, и мы отображаем первую вершину в корень, вторую вершину в правый сын корня и третью вершину в левый сын корня, такое отображение будет вложением и записываться ('', '`r`', '`l`'). С другой стороны, если мы отобразим вторую вершину на корень, а остальные две вершины на сыновья корня, это не будет допустимым вложением, так как 1 и 3 должны быть соединены ребром, а сыновья корня дерева не соединены.

По данному графу $\mathcal{G}$ и целому числу k найдите k-е в лексикографическом порядке вложение графа $\mathcal{G}$ в дерево $\mathcal{T}$. k отсчитывается от нуля.

Формат входного файла

Входной файл состоит из не более чем десяти тестовых наборов. Каждый тестовый набор начинается со строки, содержащей три целых числа n, m и k раздеMJнных одним пробелом ($1 \leqslant n \leqslant 10$, $0 \leqslant k \leqslant 1\,000\,000$). Следующие m строк описывают ребра, каждая строка содержит два целых числа u_i и v_i, раздеMJнных одним пробелом — конечные точки ребра ($1 \leqslant u_i, v_i \leqslant n$). Граф не содержит циклов и содержит не более чем одно ребро между любой парой вершин. Строка из тTJх нулей завершает ввод, этот тестовый набор не должен обрабатываться. Сумма всех k во входном файле не превышает $1\,000\,000$.

Формат выходного файла

Для каждого тестового набора выведите запись k-го (считая от нуля) вложения графа $\mathcal{G}$ в дерево $\mathcal{T}$ в лексикографическом порядке или сообщите, что такого вложения не существует. Придерживайтесь спецификации вывода, показанной на примере ниже.

Примеры

embedding.in
3 2 0 1 2 1 3 3 2 1 1 2 1 3 3 2 4 2 1 2 3 3 2 5 2 1 2 3 3 3 1 1 2 2 3 3 1 5 4 20 3 1 3 2 3 4 3 5 2 0 1 0 0 0

embedding.out
Case 1: The embedding number 0 is (‘’, ‘l’, ‘r’). Case 2: The embedding number 1 is (‘’, ‘r’, ‘l’). Case 3: The embedding number 4 is (‘l’, ‘’, ‘r’). Case 4: The embedding number 5 is (‘l’, ‘ll’, ‘lll’). Case 5: There is no such embedding. Case 6: There is no such embedding. Case 7: The embedding number 1 is (‘’, ‘ll’).

Задача D. Многопользовательский режим

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 Mebibytes

Современную компьютерную игру сложно представить без многопользовательского режима. При проработке такого режима часто возникают проблемы при организации турниров, особенно если игра ведётся «команда на команду».

Например, на сервере собралось чётное количество n игроков. Они играют турнир из нескольких дуэлей «команда из $n/2$ человек на команду из $n/2$ человек», меняя составы команд. Для того, чтобы турнир состоялся, любые два игрока должны находиться в противоположных командах как минимум дважды (дабы исключить флуктуации). Каково минимальное количество раундов может содержать состоявшийся турнир?

Например, если у нас есть 8 игроков, пронумерованных от 1 до 8, то мы можем организовать деление на команды следующим образом: $(1, 2, 3, 4)$ – $(5, 6, 7, 8)$, $(1, 2, 5, 6)$ – $(3, 4, 7, 8)$, $(1, 3, 5, 7)$ – $(2, 4, 6, 8)$, $(1, 4, 6, 7)$ – $(2, 3, 5, 8)$.

Формат входного файла

Первая строка ввода содержит количество тестов T ($1 \le T \le 50$). Каждая из следующих T строк содержит чётное число N ($2 \le N \le 300$) — количество игроков на сервере в данном тесте.

Формат выходного файла

Выведите T строк вида `Case #A: B`, где A — номер теста (начиная с 1), B — нужное количество раундов для заданного N.

Пример

standard input	standard output
4 2 4 12 16	Case #1: 2 Case #2: 3 Case #3: 5 Case #4: 5

Задача E. Life 2

Имя входного файла:	`life2.in`
Имя выходного файла:	`life2.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

RIANT (Research Institute of Analysis of Nature Transformations) discovered another strange colony of microorganisms. One famous scientist deduced recurrent formulae for $T(N)$ — living time of a colony in seconds depending on a positive integer N, the characteristic number of a colony.

$$\begin{aligned} T(1) &= 0 \\ T(2k) &= T(k) + 1 \qquad \forall k \geq 1 \\ T(2k+1) &= T(6k+4) + 1 \qquad \forall k \geq 1 \end{aligned}$$

RIANT asked Vasya to write a program which will find the living time of a colony given the characteristic number N. Since the program should work for very large values of N, Vasya needs your help.

Формат входного файла

The input file contains a single positive integer N ($N < 2^{100\,000}$). It is guaranteed that for given N, the value of $T(N)$ is not greater than 1 500 000.

Формат выходного файла

Output file must contain a single integer — $T(N)$.

Пример

life2.in	life2.out
27	111

Задача F. Сравнение картинок

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	6 секунды
Ограничение по памяти:	64 Mebibytes

Согласно основной идее, положенной в основу проекта машины, возможность объединения одновременных событий из «искусственно созданных реальностей», порождаемых различными текстами, в одной пространственной сцене зависит от того, насколько различается описание обстановки, в которой происходит действие.

Существующий уровень маготехники позволяет представить обстановку в виде чёрно-белого растрового изображения Q, состоящего из $N(Q) \times M(Q)$ пикселей (пиксели — квадраты, образованные равномерной сеткой). На изображении введена система координат: ось X направлена вниз, а ось Y — вправо.

Каждый пиксель изображения Q имеет координаты (X, Y). (X, Y) — пара целых чисел $(0 \le X < N(Q), 0 \le Y < M(Q))$. Для каждого пикселя изображения Q с координатами (i, j) задана яркость пикселя $Q[i, j]$ — целое число от 0 до 255 включительно.

Сравнение происходит следующим образом: Пусть дано два изображения: A и B. Будем говорить, что прямоугольный фрагмент изображения A с началом в (s, t) отличается от изображения B на $D[s, t]$:

$D[s,t] = \sum_{i=0...N(B)-1, j=0...M(B)-1} (A[s+i, t+j] - B[i, j])^2$

При этом s и t меняются в пределах: $0 \le s \le N(A) - N(B)$, $0 \le t \le M(A) - M(B)$.

Для решения задачи нужно найти прямоугольный фрагмент A, наименее отличающийся от изображения B. Если существует несколько таких фрагментов, то нужно выбрать фрагмент с минимальной координатой s. Если по-прежнему таких фрагментов несколько, нужно выбрать фрагмент с минимальной координатой t.

Формат входного файла

Во входном файле сначала описано изображение A, затем изображение B. Для каждого изображения Q в первой строке через пробел записаны размеры: целые числа $N(Q)$ и $M(Q)$ $(1 \le N(A), M(A) \le 500, 1 \le N(B) \le N(A), 1 \le M(B) \le M(A))$. В $N(Q)$ последующих строках записано по $M(Q)$ целых чисел через пробел. В i-ой строке j-ое число равно $Q[i, j]$ — яркость пикселя с координатами (i, j).

Формат выходного файла

В выходной файл нужно выдать координаты (s, t) начала искомого фрагмента и величину $D[s, t]$ в формате: `s t: D[s,t]`. Формат вывода предоставлен в примерах.

Пример

input.txt	output.txt
5 4 208 214 49 14 29 175 170 208 140 133 0 140 100 20 76 255 64 29 153 31 3 2 16 94 172 98 17 8	1 0: 16012
4 4 0 1 2 0 1 0 0 1 0 1 1 1 1 1 1 1 1 2 1 1	2 1: 0

Задача G. Бар “Троица”

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

Бар “Троица” — самое популярное заведение в Байтландии. Его название, вопреки мнению многих, связано не с тем, что там частенько соображают “на троих”, а с тем, что в этом баре каждый коктейль состоит равно из трех различных ингредиентов.

Хозяин бара знает, что всего у него имеется N различных ингредиентов. Из них он хочет составить максимальное количество различных коктейлей, чтобы привлечь как можно больше клиентов. Два коктейля он считает различными, если в них не больше одного совпадающего ингредиента.

Конечно, сам он этим заниматься не намерен, и поручил вам составить список коктейлей, такой чтобы выполнялись два условия:

1. Все коктейли должны быть попарно различными
2. Для любых двух имеющихся ингредиентов должен быть коктейль, в котором они присутствуют

Напишите программу, которая в зависимости от количества ингредиентов находит нужный список коктейлей, или говорит, что его не существует.

Формат входного файла

В первой строке входного файла задано единственное целое число N — количество различных ингредиентов ($3 \le N \le 300$).

Формат выходного файла

Если требуемый список коктейлей составить невозможно, то в первой строке выходного файла выведите -1. Иначе, выведите количество коктейлей в списке, а в следующих строках сами коктейли в любом порядке. Коктейль нужно выводить как три ингредиента, разделенные пробелом. Ингредиенты нумеруются от 1 до N. Если возможных ответов несколько, то выведите любой из них.

Пример

drink.in	drink.out
3	1 1 2 3
4	-1
7	7 1 2 7 1 3 6 1 4 5 2 3 5 2 4 6 3 4 7 5 6 7

Задача N. Бумеранги

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

Новейшая граната «Банзай-1» была выполнена в форме бумеранга.

Из книги «Воспоминания старого камикадзе»

— Кроме того, вы легко можете придумывать новые виды спорта. Пока участники и зрители осознают правила, а жулики ищут дырки в этих правилах, вы сможете сделать неплохие деньги, — вдохновившись, продолжал Ааз.

— Легко сказать... Вот скоро будет чемпионат по лёгкой атлетике. Допинг-тесты, допинг-чекеры, допинг-тестеры, допинг-сливы... И как тут придумать такой вид спорта, чтобы и смотрелся традиционно, и допинг там не помогал? За одну идею наша ассоциация букмекеров готова заплатить немалую сумму.

— Пожалуйста. Фигурное метание бумерангов. Правила просты: двое бросают бумеранги. Каждый бумеранг — это дуга некоторой фиксированной окружности, и он летит по этой окружности с постоянной скоростью до бесконечности (тот, кто бросил, уходит). Высший пилотаж — это пустить бумеранги так, чтобы они не столкнулись, — тут же откликнулся Ааз.

Для разработки системы тренировки атлетов Вам по заданным параметрам бумерангов требуется определить, столкнутся ли бумеранги, и если да, то когда. Бумеранги замкнутые.

Формат входного файла

Во входном файле заданы не более 10 000 тестов. Каждый тест состоит из двух бумерангов. Каждый бумеранг представлен отдельной строкой вида $x\ y\ r\ \alpha\ \beta\ o\ d$. Здесь x и y — координаты центра окружности, содержащей бумеранг ($|x|, |y| \le 100$), r — её радиус ($0 < r \le 100$), α и β — углы в градусах против часовой стрелки от оси OX, задающие начальное положение концов бумеранга на окружности ($0 \le \alpha, \beta < 360, \alpha \ne \beta$), o — ориентация дуги ('+' или '-'), а d — угловая скорость дуги в градусах в секунду ($|d| \le 360$). Ориентация '+' означает, что бумеранг — это дуга окружности от угла α до угла β против часовой стрелки, а ориентация '-' — по часовой стрелке. Знак числа d имеет аналогичный ориентации смысл — направление движения против часовой стрелки или по часовой стрелке. Файл завершается строкой из семи нулей. Все числа во входном файле целые. Изначально бумеранги не имеют общих точек.

Формат выходного файла

Для каждого из тестов выведите одну строку, содержащую ответ. Следуйте формату примера максимально точно. Ваш результат будет проверен автоматически на нахождение в границах абсолютной или относительной погрешности 10^{-6}.

Пример

boomerangs.in
0 0 1 0 90 + 10 1 0 1 0 90 + -13 0 0 1 0 90 + 10 3 0 1 0 90 + -13 0 0 0 0 0 0 0
boomerangs.out
Case 1: Collision after 64.61538461538461 seconds Case 2: They will fly infinitely! Great!