Задача A. Conspirology

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

Перед очередным всеукраинским турниром по программированию главный конспиролог — секретный агент m. — ищет, где собака зарыта.

По увиденной конспирологом во сне информации, на месте проведения контеста раньше было кладбище домашних животных. Место проведения контеста представляет собой квадрат размером $N \times N$ метров. В разных его местах агент ощущает наличие зарытой собаки, причём в разных местах интенсивность этих ощущений разная.

Квадрат разделён на N^2 квадратиков, в квадратике с координатами (i, j) mihnevsev ощущает присутствие зарытой собаки с интенсивностью $a_{i,j}$.

Квадратик с координатами $(1, 1)$ расположен в левом верхнем углу большого квадрата. Первая координата откладывается по горизонтали, вторая — по вертикали.

Вначале конспиролог находится в квадратике с координатами (i_0, j_0). На каждом шаге он перемещается в соседний (слева, сверху, справа или снизу) квадратик, в котором ощущение самое сильное и сильнее, чем в квадратике, в котором он находится в настоящий момент. Если такого квадратика не оказалось, то конспиролог останавливается.

Напишите программу, позволяющую найти траекторию конспиролога до момента остановки. Гарантируется, что траектория определяется однозначно, то есть что на пути конспиролога не будет такого квадратика, в котором дальнейшее движение нельзя определить однозначно.

Формат входного файла

Входной файл содержит целые числа N, i_0, j_0.

Далее следуют N^2 целых чисел $a_{i,j}$ ($2 \leqslant N \leqslant 100$, $0 \leqslant a_{i,j} \leqslant 100$, $1 \leqslant i_0, j_0 \leqslant N$).

Формат выходного файла

Выходной файл должен содержать пары целых чисел — траекторию конспиролога.

Пример

mihnevsev.in	mihnevsev.out
3 1 2 0 2 3 0 1 1 0 1 1	1 2 2 2 2 1 3 1

Задача B. CPR Grand Prix

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

В пробном раунде, проведённом по системе CPR на задачах чемпионата известного университета, состоящем из M задач, приняло участие N тестеров. В силу особенностей взаимоотношения тестеров с задачами за время контеста каждый тестер мог написать и «пропихнуть» любую из M задач, но ровно одну (вероятность додуматься до авторского алгоритма и решить более одной задачи на данном наборе тестеров считать равной нулю, так же как и вероятность, что никакая задача не пропихнётся).

Учитывая, что в системе CPR ценность задачи определяется количеством участников, решивших эту задачу, вычислите, какова вероятность того, что сумма полных баллов по задачам будет максимальна (т.е. что никакие два участника не сдадут одну и ту же задачу).

Формат входного файла

Входной файл содержит целые числа N и M ($1 \leqslant N \leqslant 9$, $1 \leqslant M \leqslant 9$).

Формат выходного файла

Выходной файл должен содержать вещественное число — вероятность того, что никакие два участника не сдадут одну и ту же задачу.

Ответ необходимо вывести с точностью не менее семи знаков после десятичной точки.

Примеры

cpr.in	cpr.out
2 2	0.5
2 3	0.6666666667

Задача C. Facepalm Calendar

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

Организаторы турнира Facepalm Hackers Contest решили увеличить количество неожиданностей, подстерегающих участников. Теперь ключевые даты в расписании даются в «сжатом» формате, получаемом из классического варианта “dd.mm.yy” выбрасыванием всех символов, кроме цифр от 1 до 9. Например, дата 06.10.12 в «сжатом» формате запишется как 6112. Участники сразу поняли, что одна и та же запись может соответствовать нескольким «обычным» датам

По заданной дате в «сжатом» формате выведите все даты из диапазона от 1 января 2000 г. по 31 декабря 2099 г. включительно, которые ей соответствуют.

Формат входного файла

Дата в «сжатом» формате — целое положительное число N ($1 \leqslant N \leqslant 10^6$), в записи которого отсутствуют нули.

Формат выходного файла

В первой строке выведите количество обычных дат, соответствующих заданной «сжатой» дате. В последующих строках выведите эти обычные даты в хронологическом порядке, по одной дате в строке.

Примеры

facepalm.in	facepalm.out
841	2 08.04.01 08.04.10
29244	1 29.02.44
33247	0

Задача D. Dinner after contest

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

Когда участники полуфинала, проходившего на родине легендарного вампира, собрались на обед, их ждал сюрприз: оказалось, что приготовление обеда синхронизируется с аналогичным мероприятием в одном украинском городе посредством голубиной почты.

Так что участники достали сковородку и решили пообедать купленными в магазине котлетами. Всего участникам нужно поджарить N котлет.

Каждая котлета должна быть обжарена с каждой из K сторон, одна сторона любой котлеты обжаривается в течение T минут. На сковородке помещается одновременно до P котлет. Помогите участникам определить, за какое минимальное время они смогут обжарить все котлеты со всех сторон.

Формат входного файла

Четыре натуральных числа N, K, T, P. Все числа не превосходят 1000.

Формат выходного файла

Минимальное время в минутах, необходимое, чтобы пожарить котлеты.

Примеры

seerc.in	seerc.out
1 1 5 1	5
3 1 5 2	10

Задача E. Planning for new contest

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

Организация нового турнира по программированию, который мог бы стать событием года — очень сложная задача.

Как и все большие задачи, её разбили на подзадачи. Между подзадачами есть зависимости: есть такие пары подзадач (u, v), что к подзадаче v нельзя приступать, не завершив предварительно подзадачу u. Например, не договорившись с телевизионщиками, нельзя записать интервью харизматичного преподавателя известного вуза.

При этом некоторые зависимости могут присутствовать и неявно. Например, если имеются зависимости (u, v) и (v, w), то подзадачу w нельзя выполнять, пока не будет выполнена подзадача u (например, если определение состава участников зависит от результатов отборочного раунда, а бронирование билетов зависит от состава участников, то бронирование билетов нельзя выполнять до завершения отборочного раунда).

Был составлен план организации турнира с подзадачами и зависимостями между ними. Однако выяснилось, что зависимостей в плане обозначено слишком много, так что в какой-то момент произошёл сбой — спонсор турнира запутался в организации рассылки для финалистов.

Чтобы предотвратить подобные ситуации, было принято решение некоторые зависимости из плана удалить — всё равно они будут присутствовать неявно. Вам поручили написать программу, удаляющую как можно больше лишних зависимостей.

Множество зависимостей можно удалить, если после его удаления все неявные зависимости сохранятся.

Формат входного файла

В первой строке входного файла находятся числа N, M — количество подзадач и количество зависимостей ($1 \leqslant N \leqslant 50$). Далее следует M пар чисел, обозначающих зависимости. Пара uv означает, что подзадача v не может быть выполнена, пока не будет выполнена u.

Подзадачи пронумерованы первыми N натуральными числами.

Гарантируется, что ни одна зависимость не упомянута дважды.

Гарантируется, что никакая задача не зависит от самой себя и зависимости не образуют циклов.

Формат выходного файла

Выходной файл должен содержать целое число — максимальное количество зависимостей, которые можно удалить. Затем должна следовать последовательность номеров удаляемых зависимостей, перечисленных в порядке возрастания.

Примеры

planning.in	planning.out
4 5 1 2 1 3 2 4 3 4 2 3	2 2 3

Задача F. Maximize The Quota

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

Билл очень любит число X, потому что оно приносит ему удачу. Поэтому при определении количества дополнительных путёвок в Восточную Европу в мае 2012 Билл действует следующим образом: для каждого региона он выбирает случайное T-значное число и считает, сколько раз в записи этого числа встретится число X в виде подстроки. Получившееся значение и определяет количество дополнительных путёвок в финал.

По заданным X и T вычислите среднее количество дополнительных путёвок в финал при этих параметрах.

Формат входного файла

Два целых числа: X и T ($0 \leqslant X \leqslant 10^9$, $1 \leqslant T \leqslant 10^3$).

Формат выходного файла

Выведите математическое ожидание количества дополнительных путёвок в финал для региона при заданных параметрах жеребьёвки (число X и T-значные числа) с точностью до 7 знаков после запятой.

Примеры

quota.in	quota.out
0 1	0.00
0 2	0.100000
1 3	0.3111111

Задача G. В стопицотый раз о музыкальной этике

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

Некий философ, разочаровавшийся в спортивном программировании, решил переключиться на музыку. Для этого он собрал своих единомышленников и организовал из них группу «Red or Dead». Однако собранные единомышленники оказались не особо сильны в музыке, а на просьбы объяснить хотя бы длительности нот они получали только насмешки — по мнению лидера, только так можно воспитать настоящего рок-музыканта. Так что пришлось им обращаться за помощью к вам.

Длительность ноты записывается дробью, числителем которой является число в диапазоне от 1 до 128, а знаменателем — одно из чисел $1, 2, 4, 8, 16, 32$.

Также существует модификатор длительности — символ «точка». Добавление первой точки увеличивает длительность ноты на $1/2$ её исходной длительности, добавление второй точки — ещё на $1/4$ её исходной длительности, и так далее. Например, длительность ноты с тремя точками в $15/8 = 1 + 1/2 + 1/4 + 1/8$ раз больше длительности основной ноты.

Не менее важен размер такта, который определяется как сумма длительностей всех нот, входящих в один такт. Такт — это последовательность одной или нескольких длительностей нот.

Первое задание, которое лидер группы задал своим будущим коллегам — по заданной последовательности нот в такте определить его размер. Помогите им найти ответ.

Формат входного файла

В первой строке входного файла содержится число N — количество нот в такте. В следующих N строках содержится описание нот — сначала дробь, означающая длительность ноты, затем — ноль или более точек. Числитель и знаменатель дроби разделяются символом '/' (ASCII 47). Строки с описанием нот не содержат пробелов, $1 \leqslant N \leqslant 1000$, количество точек после каждой ноты не превосходит 5.

Формат выходного файла

Выходной файл должен содержать два целых числа — числитель и знаменатель несократимой дроби, равной музыкальному размеру произведения.

Пример

rock.in	rock.out
5 1/1..... 16/32 1/2 30/32 1/32	63 16

В примере сумма длительностей нот в такте равна 1/1*(1+1/2+1/4+1/8+1/16+1/32)+16/32+1/2+30/32+1/