

Задача А. Прогнозы на голосование

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

В связи с тем, что выбор места проведения будущего финала Чемпионата Мира по программированию в последнее время стал не менее интригующей процедурой, чем сам финал, проект *VooyumRumors* объявил конкурс прогнозов на итоги голосования Оргкомитета по выбору места проведения финала. Координатору проекта удалось получить список городов-кандидатов. Остальную важную для прогноза информацию (типа высоты снежного покрова в городе во время проведения финала или взаимное расположение пальца дяди Билла и потолка в момент выбора) участники искали сами.

И вот результаты голосования (процент голосов, отданных за каждый город) известны. Осталось подвести итог конкурса, то есть выяснить, какие прогнозы оказались точными, а какие — ошибочными.

Формат входного файла

Первая строка входного файла содержит два числа p и g — количество городов, участвующих в голосовании и количество прогнозов соответственно ($1 \leq p \leq 50$, $1 \leq g \leq 10^4$). Каждая из следующих p строк содержит название города и полученный им процент голосов, указанный с одним знаком после запятой. Название каждого из городов состоит из прописных и строчных латинских букв, а также цифр, и по длине не превышает 20.

Далее идут g строк, каждая из которых описывает один прогноз. Прогноз записан в форме $P_1 + P_2 + \dots + P_k \text{ COMP } n$, где $P_1 \dots P_k$ — названия городов, COMP — один из операторов сравнения $<$, $>$, $<=$ (для \leq), $>=$ (для \geq) и $=$, а n — целое число от 0 до 100 включительно. В одном прогнозе название любого города встречается не более одного раза.

Формат выходного файла

Для каждого прогноза посчитайте сумму процентов голосов, набранных городами, и сравните их с заданным целым числом n . После этого в соответствии с форматом, заданным в примере, выведите строчку, сообщающую о том, является данный прогноз верным или же ошибочным.

Пример

<code>ballot.in</code>	<code>ballot.out</code>
6 5	Guess #1 was incorrect.
Hablin 30.7	Guess #2 was incorrect.
Mirny 20.8	Guess #3 was correct.
Sofia 12.1	Guess #4 was incorrect.
SPb 11.0	Guess #5 was correct.
E95 7.5	
Rostov 7.2	
SPb > 11	
Hablin + Mirny < 50	
Mirny + Rostov >= 28	
SPb + Mirny + Hablin <= 42	
Hablin + SPb + Mirny + E95 = 70	

Задача В. Кафе

Имя входного файла: `cafe.in`
Имя выходного файла: `cafe.out`
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 Mebibytes

В результате голосования был выбран город Хаблин в Срединной Империи, причём время проведения было выбрано таким образом, чтобы знаменитый город ледяных фигур находился в наилучшем состоянии. Однако это же обстоятельство наложило определённые требования на перемещения участников.

Учитывая, что при ожидаемой в Хаблине температуре пребывание на открытом воздухе более 10 минут становится опасным для здоровья, организаторы финала арендовали несколько кафе, расположенных на городских перекрёстках — чтобы участники по пути могли в них каким-то способом согреться. Все достопримечательности города и места проживания участников также расположены на перекрёстках. При этом каждому участнику финала, решившемуся выйти в город, должен быть предложен специальный маршрут, составленный таким образом, чтобы время участника в пути от одного арендованного кафе (или от начального перекрёстка) до другого (или до перекрёстка назначения) не превышало 10 минут, и чтобы с учётом данных требований участник дошёл до места назначения с как можно меньшим числом остановок в кафе.

Формат входного файла

Входной файл содержит несколько тестовых примеров. Каждый тестовый пример начинается со строки, содержащей целое число n ($2 \leq n \leq 10^4$) — количество перекрёстков Хаблина, при этом перекрёстки занумерованы целыми числами от 1 до n , а начальная и конечная точка пути имеют соответственно номера 1 и n . Следующая строка содержит целое число h ($0 \leq h \leq \min(n, 100)$) — количество арендованных кафе, за которым следуют числа c_1, c_2, \dots, c_h — номера перекрёстков, на которых они расположены. Следующие m строк описывают улицы Хаблина. Каждая улица задана строкой, содержащей 3 целых числа a, b, t ($1 \leq a, b \leq n, 1 \leq t \leq 600$) — соответственно номера перекрёстков, соединённых улицей и время в секундах, необходимое участнику, чтобы пройти улицу целиком. Двигаться по улице можно в обоих направлениях.

Входной файл заканчивается примером с $n = 0$, который обрабатывать не следует.

Формат выходного файла

Для каждого тестового примера выведите одну строку, содержащую наименьшее количество кафе, в которых остановится участник на пути от перекрёстка с номером 1 к перекрёстку с номером n . Если соответствующий условиям задачи маршрут не может быть найден, выведите -1 .

Пример

cafe.in	cafe.out
6	2
3 2 5 3	-1
8	
1 2 400	
3 2 80	
3 4 301	
4 5 290	
5 6 139	
1 3 375	
2 5 462	
4 6 300	
3	
0	
2	
1 2 371	
2 3 230	
0	

Задача С. Ещё один спонсор

Имя входного файла: `another.in`
Имя выходного файла: `another.out`
Ограничение по времени: 3 seconds
Ограничение по памяти: 64 Mebibytes

Секретарь Оргкома финала (в других странах эта должность называется «председатель оргкомитета», но всем известно, что в Срединной Державе Председатель один) изучал план проведения соревнований.

— Согласно европейским традициям, которые описал Ла Жеч Ни ещё в позапрошлом веке, для участников будет построен ледяной дворец, — докладывал Ли Хой, инструктор по расселению.

— Вы с ума сош, Ли! Мы и так сдвину, Ли, время финала на удобное нам, чтобы северные варвары не провели свои сборы в Гиперборее. Если мы их ещё в ледяной дом поселим... Предлагаю назначить спонсора.

— Товарищ секретарь, мы не можем ещё увеличить нагрузку на предприятия, они и так работают в три смены. Разве что кого-то из жителей города, в порядке великого почина. И пусть его укажет случай...

У жителей города Хаблин имеются облигации m серий. Серии занумерованы с единицы. Во время розыгрыша среди имеющихся на руках облигаций i -й серии случайным образом выбирается одна (причём для всех облигаций одной серии вероятность выигрыша одинакова), и её обладателю выплачивается выигрыш 2^i хуаней. Руководители Оргкома хотят, используя информацию о количестве облигаций каждого типа у жителей города, для каждого из этих жителей найти вероятность того, что после всех m розыгрышей он выиграет большую сумму, чем любой другой житель.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. Каждый тестовый пример начинается со строки, содержащей два целых числа n и m ($1 \leq n \leq 10^4$, $1 \leq m \leq 30$) — количество жителей, на руках у которых имеются облигации и количество выпущенных серий соответственно.

Каждая из следующих n строк содержит m неотрицательных целых чисел. i -е число в j -й строке обозначает количество облигаций i -й серии у j -го жителя. При этом общее количество облигаций одной серии находится в интервале от 1 до 10^9 .

Завершается входной файл строкой из двух нулей, которые обрабатывать не требуется.

Формат выходного файла

Для каждого тестового примера выведите n строк, причём в i -й строке должна содержаться вероятность того, что i -й житель выиграет наибольшее количество денег. Вероятность выводится в виде несократимой дроби (формат вывода указан в примере).

Пример

<code>another.in</code>	<code>another.out</code>
5 4	1 / 4
3 1 2 3	1 / 3
3 1 2 4	5 / 12
3 1 3 5	0 / 1
4 4 4 0	0 / 1
5 5 0 0	1 / 1
1 1	
1	
0 0	

Задача D. Тёмные улицы

Имя входного файла: `dark.in`
Имя выходного файла: `dark.out`
Ограничение по времени: 3 seconds
Ограничение по памяти: 64 Mebibytes

До финала Чемпионата Мира каждая улица в городе Хаблин была освещена. Это стоило муниципалитету 1 хуань за ли освещённой улицы в сутки. Однако в связи с необходимостью изыскания средств на возведение к финалу в знаменитом ледовом городке конной статуи дяди Билла было принято решение перейти на режим экономии и выключить освещение на некоторых улицах так, чтобы между каждыми двумя перекрёстками оставался как минимум один путь, полностью проходящий по освещённым улицам.

Определите максимальную сумму, которую муниципалитет сможет экономить ежедневно при переходе на режим экономии.

Формат входного файла

Входной файл содержит несколько тестовых примеров. Каждый тестовый пример начинается с двух чисел m и n — соответственно, количества перекрёстков и улиц в Хаблине ($1 \leq m \leq 2 \times 10^5$, $m - 1 \leq n \leq 2 \times 10^5$). Далее идут n троек целых чисел x , y , z , каждая из которых описывает двунаправленную дорогу между перекрёстками с различными номерами x и y длиной z ли ($0 \leq x, y < m$, $0 \leq z$). При этом суммарная длина всех дорог меньше, чем 2^{31} ли.

Входной файл заканчивается примером с $m = n = 0$, который обрабатывать не следует.

Формат выходного файла

Для каждого тестового примера выведите одну строку, содержащую максимальную сумму, которую сможет сэкономить муниципалитет при переходе на режим экономии.

Пример

<code>dark.in</code>	<code>dark.out</code>
7 11 0 1 7 0 3 5 1 2 8 1 3 9 1 4 7 2 4 5 3 4 15 3 5 6 4 5 8 4 6 9 5 6 11 0 0	51

Задача Е. Шведский стол

Имя входного файла: food.in
Имя выходного файла: food.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Столовой университета города Хаблин Оргкомом была поручена организация питания зарубежных участников финала.

Работники столовой решили организовать «шведский стол» с местной спецификой — любой участник, пока голоден, может бесплатно получить очередную фиксированную порцию риса. Может оказаться, что участник доест только часть последней порции — тогда остаток придётся выбросить.

Организаторы финала провели анкетирование и выяснили, сколько риса может съесть каждый участник. С целью экономии директору столовой необходимо определить оптимальный размер порции S такой, что, с одной стороны, количество выбрасываемой пищи будет достаточно мало, а с другой стороны, количество «подходов» каждого участника за добавкой также будет не очень велико.

Оптимальный размер порции вычисляется следующим образом: пусть x — количество риса, которое придётся выбросить, а y — общее количество подходов к раздаче для всех участников. Тогда размер порции выбирается с тем расчётом, чтобы сумма накладных расходов $a \times x + b \times y$ (где a — цена в хуанях одного грамма риса, а b — количество хуаней, затрачиваемых на обслуживание одного участника) была минимальной. При этом с целью предотвращения очередей ни один участник не должен делать более 3 «подходов».

Формат входного файла

Входной файл содержит несколько тестовых примеров. Каждый пример начинается со строки, содержащей целое число n ($1 \leq n \leq 1000$) — количества участников, которых необходимо обслужить. Следующая строка содержит целые числа a и b — цену в хуанях одного грамма риса и количество хуаней, затрачиваемых на обслуживание одного участника соответственно ($1 \leq a, b \leq 10$). Третья строка содержит n целых чисел y_1, y_2, \dots, y_n ($1 \leq y_i \leq 100$) — количество риса в граммах, которое может съесть каждый участник.

Входной файл завершается примером с $n = 0$, который обрабатывать не требуется.

Формат выходного файла

Для каждого тестового примера выведите одну строку, содержащую сумму накладных расходов при оптимальном выборе размера порции в виде несократимой дроби. Если результат — целое число, знаменатель 1 выводить не требуется. Более детально формат вывода указан в примере к задаче.

Пример

food.in	food.out
5	35 / 2
1 1	154 / 3
3 7 1 9 12	9
3	
10 1	
11 13 17	
2	
2 3	
6 3	
0	

Задача F. ГСЧ

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

Так как привычная для финала рассадка участников «по алфавиту» вызвала нарекания у технического персонала (алфавит Срединной Державы сильно отличается от привычной всем латиницы) Оргком предложил рассадить участников случайно.

При генерации рассадки был использован вариант метода «середин квадратов», предложенного Джоном фон Нейманом в 1946 году. Сначала выбирается начальное значение a_0 , десятичная запись которого содержит не более 4 знаков. Для получения $i + 1$ -го псевдослучайного числа a_i возводится в квадрат, запись полученного числа дополняется ведущими нулями до 8 десятичных знаков, после чего в качестве a_{i+1} берётся число, составленное из средних 4 цифр (например, при $a_0 = 1111$: $a_0^2 = 01234321$ и $a_1 = 2343$).

Для того, чтобы выполнить секретное поручение — обеспечить компактное размещение всех команд Срединной Державы — Оргкому требуется по заданному a_0 узнать, какое количество различных a_i может быть сгенерировано в результате работы данного генератора.

Формат входного файла

Входной файл содержит несколько тестовых примеров. Каждый тестовый пример состоит из одной строки, содержащей четыре цифры, задающих десятичную запись целого положительного числа a_0 , не превосходящего 10000 (если требуется, с ведущими нулями).

Входной файл завершается примером с $a_0 = 0$, который обрабатывать не требуется.

Формат выходного файла

Для каждого тестового примера выведите строку, содержащую количество различных значений a_i при заданном a_0 (включая само значение a_0).

Пример

<code>generate.in</code>	<code>generate.out</code>
5555	32
0815	17
6239	111
0	