

Задача А. Предприятие переезжает

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

На новом суперсовременном предприятии расположено N устройств, каждое из которых соединено проводами с некоторыми из M контактных щитков; каждое устройство соединено с каждым щитком не более чем одним проводом. Предприятие очень автоматизировано: информация о том, что куда подключено, хранится сразу на очень многих компьютерах. Каждый контактный щиток и каждое устройство оборудовано механическим захватом, позволяющим за одну минуту отсоединить провод от щитка или устройства, но на изменение записей потребуется еще пятьдесят девять минут. Отсоединять провода можно, только подавая команды с головного компьютера. Таким образом, после подачи команды следующую команду работы с тем же щитком или с тем же устройством можно подавать не ранее, чем через час. Более того, отсоединять провод надо одновременно с обоих концов, иначе во время, пока провод подсоединен только одним концом, в нем испортится система усиления сигнала, требующая электропитания с обоих концов провода.

Сейчас предприятие переезжает в новое здание, поэтому необходимо отсоединить все устройства от всех контактных щитков, и как можно быстрее. Таким образом, необходимо определить, когда должен отсоединяться какой провод. Напишите программу, которая решит эту задачу.

Формат входного файла

Первая строка входного файла содержит два числа N и M — количество устройств и контактных щитков ($1 \leq M, N \leq 300$). В i -й из следующих N строк содержится информация о том, к каким щиткам подключено i -е устройство: первое число в этой строке — M_i — число проводов, выходящих из i -го устройства, а далее следуют M_i чисел, задающих номера щитков, к которым подключены эти провода.

Формат выходного файла

В выходной файл выведите N строк, в i -ю из которых выведите M_i целых чисел — время в часах, когда нужно начать отсоединение соответствующих проводов.

Начальный момент времени — 0 часов.

Пример

<code>move.in</code>	<code>move.out</code>
2 3	1 0 2
3 1 2 3	0
1 3	

Задача В. Редукция дерева

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Задано неориентированное дерево, содержащее n вершин. Можно выбрать некоторое ребро и удалить его, при этом инцидентные ему вершины не удаляются. Таким образом можно удалить из дерева некоторый набор ребер. В результате дерево распадается на некоторое количество меньших деревьев. Требуется, удалив наименьшее количество ребер, получить в качестве хотя бы одной из компонент связности дерево, содержащее ровно p вершин.

Формат входного файла

Первая строка входного файла содержит пару натуральных чисел n, p , ($1 \leq p \leq n \leq 250$). Далее в $n - 1$ строке содержатся описания ребер дерева. Каждое описание состоит из пары натуральных чисел a_i, b_i ($1 \leq a_i, b_i \leq n$) — номеров соединяемых ребром вершин.

Формат выходного файла

В первую строку выведите наименьшее количество ребер q в искомом наборе. Во вторую строку выведите номера удаляемых ребер. Номера ребер определяются порядком их задания по входному файлу. Если оптимальных решений несколько, разрешается выводить любое.

Пример

<code>tree.in</code>	<code>tree.out</code>
11 6	2
1 2	3 6
1 3	
1 4	
2 6	
2 7	
1 5	
2 8	
4 9	
4 10	
4 11	

Задача С. Космический покер 3

Имя входного файла: poker.in
Имя выходного файла: poker.out
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Космический покер. Легендарная игра, первая версия которой появилась ещё в далёком 1284 году Чужой эры. До сих пор её правила известны лишь узкому кругу профессиональных игроков. Однако Вам повезло — разработчики первой в мире программы, играющей в космический покер, обратились к Вам за помощью.

В космический покер играют N инопланетян. В начале раунда каждому из игроков раздают по M карт (назовём их *личными*). Личные карты игрока неизвестны его соперникам. Затем на стол по очереди выкладываются K *общих* карт. Общие карты кладут рубашкой вниз, поэтому они видны всем игрокам. *Рука* игрока состоит из его личных и общих карт — итого $M + K$ карт. Мастей нет, карты различаются лишь достоинством. Всего есть 13 различных достоинств: 2, 3, 4, ..., 9, T, J, Q, K и A. **Играют бесконечной колодой, в которой вероятность того, что очередная карта будет иметь заданное достоинство, равна $1/13$.** Комбинации в космическом покере имеют вид (v_1, \dots, v_L) , где L — количество различных достоинств в комбинации. Рука игрока удовлетворяет комбинации (v_1, \dots, v_L) , если содержит v_1 карт первого достоинства, v_2 карт второго достоинства, ..., v_L карт L -го достоинства. Например, комбинации (2, 2) удовлетворяют руки 2JA2A и 22233. Комбинации (2, 3) удовлетворяет рука KQKQKQ, но не удовлетворяет рука AAAAAA. Все комбинации имеют различную стоимость. В раунде побеждает игрок, рука которого содержит комбинацию наибольшей стоимости среди всех комбинаций на руках всех игроков. Если таких игроков несколько, объявляется ничья.

Зная личные карты первого игрока и частично открытые общие карты, посчитайте вероятность того, что этот игрок окажется единоличным победителем раунда.

Формат входного файла

В первой строке через пробел записаны числа N , M и K ($2 \leq N, M \leq 10$; $1 \leq K \leq 5$). Во второй строке записаны M символов — личные карты первого игрока. В третьей строке записано не более K символов — открытые общие карты. В четвёртой строке записано число C — количество существующих в космическом покере комбинаций ($1 \leq C \leq 100$). Далее в C строках перечислены комбинации в порядке возрастания стоимости. Каждая из них имеет вид $L v_1 v_2 \dots v_L$. Числа L и v_i положительны, сумма всех v_i не превышает $M + K$.

Формат выходного файла

Выведите вероятность победы первого игрока с точностью не менее 10^{-5} .

Примеры

poker.in	poker.out
2 5 2 23456 1 1 2	0.0883526857
2 5 2 23456 78 2 7 1 1 1 1 1 1 1 1 4	0.8407915043