

Задача А. Яблочное дерево

Имя входного файла:	стандартный поток ввода
Имя выходного файла:	стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Валера очень любит есть яблоки и, на его счастье, около дома растет большая яблоня. Валера высокий, но яблоня росла на каком-то удивительном удобрении и выросла огромной, поэтому Валера не всегда может дотянуться до яблока. В один прекрасный день ему надоела эта несправедливость, и он решил написать программу, определяющую, какое яблоко расположено ближе всех к земле.

Яблоня представляет собой набор узлов, соединенных ветками. Из каждого узла либо растут ветки (одна или несколько), либо в этом узле находится яблоко. Развитие яблони начиналось из корневого узла. Узлы яблони пронумерованы числами от 1 до N таким образом, что корень имеет номер 1. Будем считать, что высота, на которой растет яблоко, равна расстоянию (измеренному в количестве веток) от узла, на котором растет это яблоко, до корневого узла.

Всё было бы просто, но яблоня — живой организм, и с течением времени ее структура меняется: появляются новые ветки, отваливаются старые узлы (вместе со всеми ветками и узлами, которые на них держались). Валеру постоянно интересует информация о самом нижнем яблоке, поэтому он хочет получать её после каждого изменения в структуре яблони.

Формат входных данных

В первой строке ввода содержится число узлов яблони N ($1 \leq N \leq 1,5 \times 10^5$). В следующих $N - 1$ строках содержится описание веток яблони. Каждая ветка описывается двумя целыми числами u_i, v_i , обозначающими номера узлов, которые она соединяет ($1 \leq i < N, 1 \leq u_i, v_i \leq N$). Гарантируется, что ветки не образуют циклов, а любой узел соединен (напрямую или через другие узлы) ветками с корнем дерева.

В следующей строке содержится целое число T — число событий в «жизни» яблони ($1 \leq T \leq 1,5 \times 10^5$). Далее идет T строк с описанием событий:

- $1 \ v \ u$ — в узле v выросла новая ветка. Узлу, в котором она заканчивается (а это получается новый узел) присваивается номер u . В новом узле u сразу же вырастает яблоко. Если при этом в узле v было яблоко, то оно упало и больше никого не интересует.
- $2 \ v$ — узел v отделяется от яблони. Вместе с узлом v отделяется и все, что непосредственно или косвенно за него держалось. Если в узле, за который держался узел v , не осталось веток (кроме идущей к корню), то там вырастает яблоко.

Гарантируется, что все события корректны, то есть узел с номером v есть в дереве, узла с номером u нет в дереве, а корень дерева никогда не будет отделен. Все числа положительны и не превосходят $1,5 \times 10^5$.

Обратите внимание, что если какой-то узел оказывается отделен от дерева, то его номер может быть потом использован для обозначения нового узла.

Формат выходных данных

После каждого события необходимо вывести в отдельной строке единственное число — номер наиболее близкого к корню узла, в котором есть яблоко. Если ответов несколько, то можно вывести любой.

Примеры

стандартный поток ввода	стандартный поток вывода
4	4
1 3	3
1 4	2
2 3	2
6	3
2 2	3
1 4 2	
1 3 5	
1 5 6	
2 5	
2 2	

Задача В. DCBA

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В Берляндии на уроке программирования Вольдемар попросили решить следующую задачу. Дано натуральное число N , необходимо подобрать четыре различных натуральных числа A, B, C, D такие, что $A + B = C \times D = N$.

Вольдемар ленится и просит вас помочь ему.

Формат входных данных

В первой строке ввода задано одно натуральное число N ($5 \leq N \leq 1042$).

Формат выходных данных

Выведите четыре различных натуральных числа A, B, C, D таких, что $A + B = C \times D = N$. Каждое из чисел не должно превышать N . Если решений несколько, то разрешается вывести любое из них.

Примеры

стандартный поток ввода	стандартный поток вывода
15	7 8 3 5
42	4 38 3 14

Задача С. Популяция слонов

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Академик Григорий изучает изменение численности популяции слонов с течением времени. Он заметил следующую интересную закономерность. Популяция слонов в n -ом году выражается функцией $f(n) = f(f(n-1)) + f(n - f(n-1))$, причем $f(1) = 1$, $f(2) = 2$.

Григорий просит вас посчитать количество слонов в популяции в году с заданным номером N .

Формат входных данных

В единственной строке дано число N — интересующий Григория год ($1 \leq N \leq 50$).

Формат выходных данных

В единственную строку выведите численность популяции слонов в N -ом году.

Примеры

стандартный поток ввода	стандартный поток вывода
1	1
2	2

Задача D. Красивые даты

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Юный Вася очень любит красивые даты. По мнению Васи, дата в формате год/месяц/день считается красивой, если номер года строго больше номера месяца, а номер месяца строго больше номера дня. Например, 2014/10/9 и 1992/12/11 — красивые даты, а 1/2/3 и 3/2/4 — нет.

У Васи есть любимое число k , и он хочет найти k -ю по порядку красивую дату. Так как Вася ещё слишком юн, он попросил вас помочь ему. Первой красивой датой считается дата 3/2/1

Напомним, что дата $y_1/m_1/d_1$ идёт по порядку раньше даты $y_2/m_2/d_2$, если либо $y_1 < y_2$, либо $y_1 = y_2$ и $m_1 < m_2$, либо $y_1 = y_2$, $m_1 = m_2$ и $d_1 < d_2$.

Формат входных данных

В единственной строке дано число k — любимое число Васи ($1 \leq k \leq 10^9$).

Формат выходных данных

В единственную строку выведите дату в формате год/месяц/день — k -ую по порядку красивую дату.

Примеры

стандартный поток ввода	стандартный поток вывода
1	3/2/1
2	4/2/1
3	4/3/2
130900	1992/12/11

Примечание

Обратите внимание, что при выводе даты не надо выводить лидирующие нули в описании месяца и дня.

Задача Е. Энугольник

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Если вы один из тех гиков, которые собираются по субботам, заказывают пиццу и обсуждают проблемы (а может и задачи), с которыми столкнулись за последнюю неделю, то вы наверное знакомы с тем интересным фактом, что круглые пиццы кладут в квадратные коробки и режут на треугольные части.

Чтобы подчеркнуть эту особенность, одна фирма, занимающаяся изготовлением коробок для пиццы, решила выпускать коробки новой формы. Коробки новой формы должны представлять собой выпуклые n -угольники.

Вас просят по заданному радиусу пиццы r и количеству углов коробки n выдать любую подходящую форму коробки. Иными словами, вы должны выдать выпуклый n -угольник такой, что в него вписана окружность радиуса r с центром в начале координат.

Формат входных данных

В первой строке ввода содержатся два целых числа n, r — количество углов предполагаемой коробки и радиус пиццы ($3 \leq n \leq 50, 1 \leq r \leq 10000$).

Формат выходных данных

Выведите n пар вещественных чисел x_i, y_i ($|x_i|, |y_i| \leq 10^8$), которые описывают выпуклый n -угольник в порядке обхода вершин против часовой стрелки. При этом никакие три вершины не должны лежать на одной прямой. При проверке ответа числа будут сравниваться с погрешностью 10^{-6} по абсолютному значению.

Примеры

стандартный поток ввода	стандартный поток вывода
3 3	10.00000000 0.00000000 -3.00000000 6.00000000 -3.00000000 -6.00000000

Примечание

Выпуклым многоугольником называется многоугольник, все точки которого лежат по одну сторону от любой прямой, проходящей через две его соседние вершины.

Окружность называется вписанной в n -угольник, если она касается всех его сторон.

Задача F. Робот в серверной

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вы — системный администратор одной очень перспективной IT-фирмы. В чём именно заключаются перспективы этой фирмы никто не знает, но все уверены, что они есть. Под вашим управлением находится серверная, в которой установлено целых n серверов (или n целых серверов, вы точно не уверены), пронумерованных целыми числами от 1 до n .

Ваша работа — следить за тем, чтобы серверы работали и не выключались. Если какой-нибудь сервер выключится, вы должны моментально его включить (по крайней мере, так считает ваш начальник). Вашу работу усложняет тот факт, что работоспособность некоторых серверов зависит от других серверов (например, для работы Web-сервера должен работать сервер баз данных). Если попытаться включить сервер, когда работают не все сервера, от которых он зависит, то этот сервер просто не включится (и потом придется пытаться включать его заново). Например, если сервер номер 1 зависит от серверов 2, 3 и 5, то для того, чтобы его включить, требуется сначала включить серверы 2, 3 и 5.

Как истинный системный администратор, вы ленивы и поэтому решили переложить эту сложную задачу на плечи маленького ни в чём не повинного робота. Бедняга робот не смог вам отказать, поэтому каждый день он выполняет следующий алгоритм. У робота в памяти записана последовательность из n различных чисел от 1 до n — порядок, в котором он обходит серверы. Подходя к очередному серверу, робот проверяет, выключен ли сервер, и если он выключен, то пытается его включить. После того, как робот дошёл до последнего сервера, он заканчивает выполнение алгоритма.

Недавно в серверной отключилось электричество, поэтому сейчас все серверы выключены. Вам интересно — сколько раз роботу придется выполнить свой алгоритм, чтобы все серверы оказались включенными.

Формат входных данных

Первая строка ввода содержит число n — количество серверов в серверной ($1 \leq n \leq 10^5$). Следующая строка содержит n различных чисел от 1 до n — порядок, в котором робот обходит серверы. В следующих n строках задано описание зависимостей между серверами: i -я строка содержит число k_i — количество серверов, от которых зависит i -й сервер, и последовательность k_i различных чисел — номера этих серверов. Сервер не может зависеть от самого себя. Суммарное количество зависимостей не превышает 1 000 000.

Формат выходных данных

В единственную строку выведите количество итераций алгоритма, через которое робот включит все серверы, или -1 , если робот так и не сможет включить все серверы.

Примеры

стандартный поток ввода	стандартный поток вывода
3 3 1 2 1 2 0 0	2
3 2 1 3 1 2 1 3 1 1	-1

Примечание

Подробное описание действий в первом примере:

1. Робот подходит к серверу 3 и включает его, так как этот сервер не зависит от других серверов;
2. Робот подходит к серверу 1 и не включает его, так как ещё не включен сервер 2, от которого зависит сервер 1;
3. Робот подходит к серверу 2 и включает его, так как этот сервер не зависит от других серверов;
4. Заканчивается первая итерация;
5. Робот подходит к серверу 3, сервер уже включен;
6. Робот подходит к серверу 1 и включает его, так как все серверы, от которых он зависит, уже включены;
7. Робот подходит к серверу 2, сервер уже включен;
8. Все серверы включены, прошло две итерации;

Во втором примере невозможно включить все серверы.

Задача G. Tesla Model S

Имя входного файла:	стандартный поток ввода
Имя выходного файла:	стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Джон купил новый автомобиль — **Tesla Model S**. Как известно, этот автомобиль надо заряжать на специальных заправочных станциях, которые называются «**SuperCharger**». Такая станция позволяет зарядить автомобиль **Tesla** за 20 минут — что не сильно отличается от времени заправки обычного автомобиля.

Джон, как истинный американец, любит путешествовать на своей машине по стране. Однако сеть заправок **SuperCharger** пока развита не так сильно, и иногда путешествие из одного города в другой с использованием таких заправок невозможно.

Джон составил карту автодорог США, отметил на ней заправки **SuperCharger** и теперь интересуется, можно ли из одного заданного города доехать в другой, возможно, используя заправки **SuperCharger**. Его не интересует кратчайший путь — любой путь, позволяющий проехать с использованием заправок **SuperCharger**, его устроит. Изначально машина Джона полностью заряжена.

Формат входных данных

В первой строке ввода находятся четыре числа N , M , K , P — количество городов, количество дорог между городами, количество городов, в которых есть заправка **SuperCharger** и расстояние, которое **Tesla Model S** может проехать на полностью заряженном аккумуляторе ($1 \leq N \leq 10^5, 1 \leq M \leq 3 \cdot 10^5, 1 \leq K \leq N, 1 \leq P \leq 10^9$).

Во второй строке находится K чисел — номера городов, в которых есть заправка **SuperCharger**. Города нумеруются с единицы.

Далее следует M строк — описания дорог на карте. Каждая дорога описывается тремя числами a_i, b_i, c_i — номера городов, которые она соединяет и длина дороги ($1 \leq a_i, b_i \leq N, 1 \leq c_i \leq 10^9$).

Город, в котором Джон начинает путешествие, имеет номер 1, город, в котором он финиширует, имеет номер N . В начале путешествия машина Джона полностью заряжена.

Формат выходных данных

В первой строке выведите число T — количество городов, которые потребуется посетить Джону, чтобы доехать из города 1 в город N . Во второй строке выведите номера этих городов в порядке посещения. Помните, что Джон не требует найти кратчайший путь, однако расстояние, пройденное между дозаправками, должно быть не более P . Если возможных путей несколько, выведите любой из них. Изначально автомобиль Джона полностью заряжен. В пути, который вы выведете, не должно быть больше, чем $3 \cdot 10^6$ вершин.

Если доехать из города 1 в город N невозможно, выведите единственное число -1 .

Примеры

стандартный поток ввода	стандартный поток вывода
4 4 1 10 2 1 4 11 1 2 9 2 3 5 3 4 5	4 1 2 3 4
6 7 3 5 1 2 3 1 2 1 2 3 1 3 1 1 3 4 4 4 5 1 5 6 1 4 6 2	-1
3 3 0 3 1 2 1 2 3 1 1 3 1	3 1 2 3

В первом примере дорога из города 1 в город 4 слишком длинная, чтобы проехать её на одной заправке, поэтому Джону приходится ехать в объезд.

Во втором примере Джон может доехать до городов 1, 2, 3, 4, 5, но на дорогу до последнего города ему не хватает заряда.

Третий пример показывает, что кратчайший путь искать необязательно.

Задача Н. Гонки

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На Берляндском чемпионате по парным гонкам кипели нешуточные страсти. К финальному заезду накал страстей был таков, что компьютер жюри не выдержал и перегрелся. Теперь у жюри остались лишь перемешанные записи о том, какие пары участников встречались в одном заезде.

Чемпионат проходил по следующим правилам. К заезду допускаются два участника. Проигравший спортсмен выбывает из чемпионата и больше участвовать не может. Заезды проводятся до тех пор, пока есть более одного не выбывшего участника. Каждый заявленный спортсмен участвует хотя бы в одном заезде.

Чтобы опубликовать итоговую информацию о прошедшем чемпионате на своем сайте, жюри наняло вас, сотрудника «БерляндПрограмминг», для восстановления истории заездов. Поскольку жюри понимает, что не во всех случаях можно однозначно восстановить историю, вас просят восстановить любую историю заездов, не противоречащую правилам, состоящую из оставшегося у жюри списка заездов, либо определить, что в предоставленные данные закралась ошибка.

Формат входных данных

В первой строке ввода задано два целых числа N и M ($1 \leq N, M \leq 100000$) — количество заездов и количество участников, соответственно. В следующих N строках задано два целых числа i и j ($1 \leq i, j \leq M, i \neq j$) — номера участников в заезде.

Формат выходных данных

Выведите N строк по два целых числа в каждой — номера участников очередного заезда. Все пары чисел должны повторять список заездов из ввода. Порядок участников внутри пары может быть произвольным. Выведенный порядок заездов должен соответствовать правилам чемпионата. В случае, если существует несколько ответов, разрешается вывести любой из них.

Если восстановить историю заездов, не противоречащую правилам, по заданному списку невозможно, выведите фразу «INVALID DATA» без кавычек.

Примеры

стандартный поток ввода	стандартный поток вывода
3 4 1 2 1 4 3 4	1 2 3 4 1 4
3 4 1 2 1 3 1 4	1 4 2 1 3 1
5 6 2 1 2 3 4 5 5 6 4 6	INVALID DATA

Задача I. Распознавание смайликов

Имя входного файла: стандартный поток ввода
Имя выходного файла: стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ваш друг — начинающий хакер. Чтобы помочь вам выиграть «Открытый чемпионат ПетрГУ по программированию», он взломал супер-безопасный мессенджер Telefon и получил отрывок из истории чата жюри олимпиады, которую сразу же переслал вам.

Nikita Ioffe: Я придумал простую задачу на олимпиаду.

Denis Denisov: Рассказывай.

Nikita Ioffe: Дан смайлик, нарисованный ASCII графикой, определить, весёлый он или грустный.

Alexander Fedulin: Норм. И решение писать не надо, только тесты подготовить.

Vladimir Basunkov: Прикольная задачка, а как решать?

Alexander Fedulin: Да какая разница? Пусть участники думают.

Valera Filev: Мне нравится задачка, и решается просто.

Vladimir Basunkov: Рассказывай.

Valera Filev: невозможно получить текст, так как файл повреждён.

Valera Filev: невозможно получить текст, так как файл повреждён.

Valera Filev: невозможно получить текст, так как файл повреждён.

Valera Filev: невозможно получить текст, так как файл повреждён.

Alexander Fedulin: Круто. И кода должно получиться немного, всего строк 500.

Nikita Ioffe: Я уже тесты сделал, зацените. <http://acm.petrstu.ru/files/smileys.zip>

Vladimir Basunkov: Смайлики зачёт! (y)

Как вы могли заметить, в процессе передачи файл оказался повреждён. Хватит ли вам информации, которую вы получили, чтобы решить эту простую задачу?

Формат входных данных

В первой строке входного файла находятся два числа h, w — размеры изображения со смайликом ($10 \leq h, w \leq 80$). В следующих h строках, по w символов в каждой, содержится описание самого изображения. Смайлик состоит из символов x.

Формат выходных данных

В единственную строку выходного файла выведите строку `happy`, если смайлик счастливый, и строку `sad` иначе.

Примеры

стандартный поток ввода	стандартный поток вывода
Пример находится в файле 001	happy
Пример находится в файле 002	sad

Примечание

Тесты, на которых будет проверяться ваше решение, находятся по адресу <http://acm.petrstu.ru/files/smileys.zip>.