

Задача А. Паросочетание

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Двудольным графом называется неориентированный граф (V, E) , $E \subseteq V \times V$ такой, что его множество вершин V можно разбить на два множества A и B , для которых $\forall (e_1, e_2) \in E$ $e_1 \in A$, $e_2 \in B$ и $A \cup B = V$, $A \cap B = \emptyset$.

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор $S \subseteq E$, что для любых двух рёбер $e_1 = (u_1, v_1)$, $e_2 = (u_2, v_2)$ из S $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 250$), где n — число вершин в множестве A , а m — число вершин в B .

Далее следуют n строк с описаниями рёбер — i -я вершина из A описана в $(i + 1)$ -й строке файла. Каждая из этих строк содержит номера вершин из B , соединённых с i -й вершиной A . Гарантируется, что в графе нет кратных ребер. Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

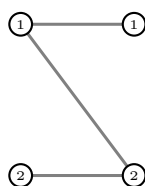
Формат выходных данных

Первая строка выходного файла должна содержать одно целое число l — количество рёбер в максимальном паросочетании. Далее следуют l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы рёбер паросочетания в A и B соответственно.

Пример

стандартный ввод	стандартный вывод
2 2	2
1 2 0	1 1
2 0	2 2

Замечание



Задача В. Покрытие путями

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

64 мегабайта

Задан ориентированный ациклический граф. Требуется определить минимальное количество не пересекающихся по вершинам путей, покрывающих все вершины.

Формат входных данных

Первая строка входного файла содержит целые числа n и m — количества вершин и рёбер графа соответственно ($2 \leq n \leq 1000$, $0 \leq m \leq 10^5$). В следующих m строках содержатся по два натуральных числа — номера вершин u и v , которые соединяет ребро (u, v) .

Формат выходных данных

В первой строке выходного файла выведите натуральное число k — минимальное количество путей, необходимых для покрытия всех вершин.

Пример

стандартный ввод	стандартный вывод
3 3 1 3 3 2 1 2	1

Задача С. День рождения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \leq k \leq 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \leq m \leq 150$ и $0 \leq n \leq 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i -й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Формат выходных данных

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Если максимальных наборов несколько, то выведите любой из них.

Пример

стандартный ввод	стандартный вывод
2	4
2 2	2 2
1 2 0	1 2
1 2 0	1 2
3 2	4
1 2 0	2 2
2 0	1 3
1 2 0	1 2

Задача D. Проблема падишаха

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Мудрый падишах внимательно следит за благополучием своих подданных, когда вершит их судьбы. В частности, на нем все заботы о вступающих в брачный возраст юношах и девушках его страны. И, как положено серьезному правителю, все по науке — перед тем, как творить молодые семьи, падишах провел Глобальное тестирование и по 100-балльной шкале определил совместимость всех юношей и девушек в совместном браке.

А дальше что? Падишах наслышан про задачу о назначении, но ему не нравится ее установка. Действительно, может ли быть спокойна его душа даже в случае всеобщего благополучия, если кому-то из подданных плохо? И можно ли жертвовать интересами хотя бы одной семьи во благо общества? Конечно, нет!

Падишаху милее другая мысль. Он хочет создать максимальное число семей, причем сделать это таким образом, чтобы минимальная совместимость в семье была максимальной. А решить эту неклассическую задачу он просит вас. Помогите падишаху!

Формат входных данных

В первой строке входных данных содержатся два целых числа n и m — количество юношей и количество девушек соответственно ($1 \leq n, m \leq 200$). Последующие n строк содержат по m целых чисел от 0 до 10^9 — коэффициент совместимости соответствующей пары (меньшее значение менее способствует супружеской жизни).

Формат выходных данных

В единственную строку выходного файла выведите наименьший искомый балл, при котором возможно создание максимально возможного количества семейных пар.

Пример

стандартный ввод	стандартный вывод
3 4 77 88 31 67 96 30 2 68 35 39 76 45	76

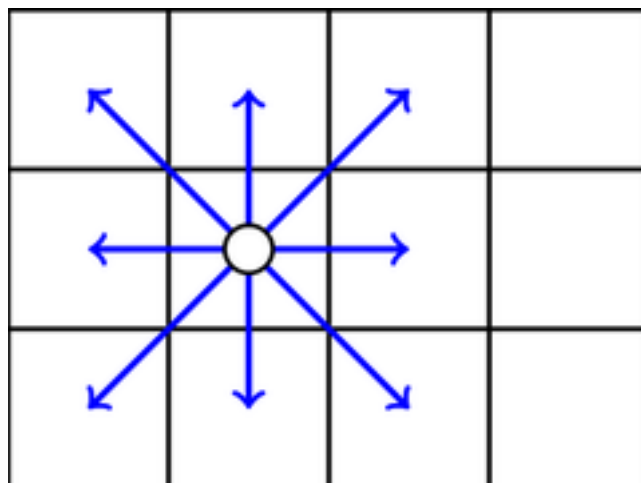
Задача E. Полезные ископаемые

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ведется проект по освоению планеты соседней звездной системы. Для добычи полезных ископаемых планируется направить на планету несколько партий роботов.

Участок поверхности планеты, на котором планируется добывать полезные ископаемые, представляет собой клетчатый прямоугольник размером w на h , клетки участка имеют координаты от $(1, 1)$ до (w, h) . В некоторых клетках участка находятся базы специалистов, в которые могут быть доставлены партии роботов. Всего на участке размещено s баз, и i -я база находится в клетке с координатами (x_i, y_i) .

Каждая партия роботов характеризуется тремя параметрами: j -я партия доставляется на базу b_j , содержит n_j роботов и каждый робот партии обладает мобильностью m_j . Когда партия роботов доставляется на соответствующую базу, каждый робот этой партии перемещается по поверхности планеты от базы до некоторой клетки. Если мобильность робота равна m , он может не более m раз переместиться на одну из восьми соседних клеток, как показано на рис. 1.



После того как роботы из всех доставленных партий размещаются на участке, они активируются и начинают добычу полезных ископаемых. В процессе перемещения в одной клетке может одновременно находиться произвольное количество роботов. Однако после активации в каждой клетке должно находиться не более q роботов.

Руководством проекта получена информация о t партиях роботов, которые могут быть последовательно отправлены на планету. После доставки всех партий роботов, учитывая их ограниченную мобильность, возможна ситуация, что не удастся разместить роботов на участке так, чтобы в каждой клетке оказалось не больше q роботов. Поэтому руководство должно выбрать k первых партий роботов, где $0 \leq k \leq t$, которые будут полностью доставлены на соответствующие базы. После этого, если $k < t$, следует дополнительно принять z из n_{k+1} роботов следующей, $(k + 1)$ -й партии, $0 \leq z < n_{k+1}$.

Все полученные таким образом роботы должны с учетом ограничения на мобильность разместиться на участке таким образом, чтобы в каждой клетке было не более q роботов. После этого они будут активированы и начнут добычу полезных ископаемых. Разумеется, руководство проекта старается максимизировать количество роботов, которые будут доставлены на планету, поэтому, с учетом описанных ограничений, требуется максимизировать k , а затем максимизировать z .

Требуется написать программу, которая по размерам участка, числу q , описанию расположения баз, а также количеству запланированных партий роботов и их описанию определяет максимальное число k — количество партий роботов, и затем — максимальное число z — дополнительное количество роботов из $(k + 1)$ -й партии, чтобы, доставив роботов на планету, их можно было разместить на участке таким образом, чтобы в каждой клетке оказалось не более q роботов.

Формат входных данных

Первая строка входного файла содержит числа w, h, s и q ($1 \leq w, h \leq 10^5, 1 \leq s \leq 4, 1 \leq q \leq 100$). Последующие s строк содержат по два целых числа x_i, y_i и описывают базы специалистов ($1 \leq x_i \leq w, 1 \leq y_i \leq h$).

Следующая строка содержит число t – количество партий роботов ($1 \leq t \leq 100$). Последующие t строк описывают партии роботов и содержат по 3 целых числа: b_j, n_j и m_j ($1 \leq b_j \leq s, 1 \leq n_j \leq w \cdot h \cdot q, 0 \leq m_j < \max(w, h)$).

Формат выходных данных

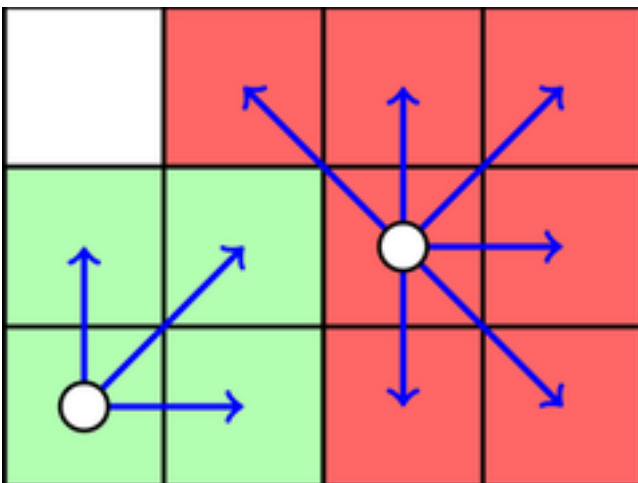
Требуется вывести два числа: k и $z, 0 \leq k \leq t$. Если $k = t$, то z должно быть равно 0, иначе должно выполняться условие $0 \leq z < n_{k+1}$.

Пример

стандартный ввод	стандартный вывод
4 3 2 1 1 1 3 2 3 1 4 1 2 9 1 1 12 2	1 7

Замечание

В приведенном примере описания входных данных следует полностью принять первую партию роботов и дополнительно принять 7 роботов из второй партии. На рис. 2 показано, как можно разместить этих роботов на участке, чтобы в каждой клетке было не более одного робота. Базы специалистов показаны кружками. Клетки, в которых окажутся роботы с базы 1, показаны зелёным, а клетки, в которых окажутся роботы с базы 2, показаны красным цветом.



Возможное размещение роботов на участке в данном примере.

Задача F. Минимальное контролирующее множество

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

Формат входных данных

В первой строке файла даны два числа m и n ($1 \leq m, n \leq 4000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leq K_i \leq n$) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входном файле не превосходит 500 000. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе — m чисел $0 \leq L_i \leq n$ — соответствующая i -й вершине первой доли вершина второй доли, или 0, если i -я вершина первой доли не входит в паросочетание.

Формат выходных данных

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли S , после которого записаны S чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Примеры

стандартный ввод	стандартный вывод
3 2	2
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

Задача G. Замощение доминошками

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано игровое поле размера $n \times m$, некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера 1×2 стоит a условных единиц, а замостить свободную клетку поля квадратиком размера 1×1 — b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

Формат входных данных

Первая строка входного файла содержит 4 целых числа n, m, a, b ($1 \leq n, m \leq 100, |a| \leq 1000, |b| \leq 1000$). Каждая из последующих n строк содержит по m символов: символ '.' (точка) обозначает занятую клетку поля, а символ '*' (звёздочка) — свободную.

Формат выходных данных

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

Пример

стандартный ввод	стандартный вывод
2 3 3 2 .** .*.	5

Задача N. Студенты в лесу

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Однажды Иван Смирнов пообещал всех студентов, которые не ходят на пары, увезти с завязанными глазами в лес. И вот он выполнил свое обещание и вывез всех прогульчиков в лес, который представляет клетчатое поле $n \times m$. Вам выдана карта этого леса, клетки, занятые деревьями, обозначены символом «#», а свободные клетки — символом «.».

К счастью студентов их преподаватель не был извергом, и в этом лесу вполне себе можно было выживать. Более того, в лесу был бесплатный Wi-Fi, поэтому все студенты решили написать раунд на `codeforces`. Все студенты ВШЭ очень честные, а поэтому не могут писать раунд, находясь в соседних клетках.

Вам требуется рассадить наибольшее количество студентов в лесу так, чтобы никакие два студента не находились в соседних клетках.

Формат входных данных

В первой строке задана пара чисел n и m ($1 \leq n, m \leq 100$).

В следующих n строках содержатся строки, по m символов каждая, описывающие лес.

Формат выходных данных

Выведите поле $n \times m$ с максимальным количеством студентов. Студенты обозначаются символом «A».

Примеры

стандартный ввод	стандартный вывод
3 6 #.##.# #.##.#	#A##A# A.A.A #A##A#
3 3	A.A .A. A.A

Задача I. Yesterday

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Мало кто знает, но Пол Маккартни помимо сочинения музыки увлекался нумерологией. Больше всего Полу нравились числа от 0 до $2^n - 1$, которые при записи в виде n -битных чисел содержали одинаковое количество нулей и единиц. Пусть x_0, \dots, x_{m-1} — это такие числа, записанные в возрастающем порядке.

На одно из дней рождений Полу подарили перестановку π_0, \dots, π_{m-1} ($0 \leq \pi_i < m$, все π_i различны) со следующим замечательным свойством: для любого i числа x_i и x_{π_i} различаются в двоичной записи ровно в k битах. Пол очень любил эту перестановку, носил ее с собой на репетиции, гулял с ней и бла-бла-бла.

Потом Пол потерял перестановку, сильно расстроился и написал песню Yesterday. В то время не было ни производительных ЭВМ, и было придумано не так уж и много алгоритмов, так что новую перестановку с такими же свойствами найти никто не смог.

Но сейчас все это есть, так что у вас нет никаких оправданий, чтобы не найти такую перестановку!

Формат входных данных

В единственной строке указана пара чисел n и k ($2 \leq n \leq 20$, $0 \leq k \leq n$, n — четное).

Формат выходных данных

Выведите подходящую перестановку, если такая существует, или -1 , в ином случае.

Примеры

стандартный ввод	стандартный вывод
4 0	0 1 2 3 4 5
4 1	-1
4 2	4 3 1 5 0 2

Замечание

При $n = 4$ массив x выглядит следующим образом: $x = [3, 5, 6, 9, 10, 13]$.

Задача J. Спонсоры

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Чернорусы решили провести первую в мире чернорусскую олимпиаду (ЧРО). Чтобы провести такое масштабное мероприятие, они прибегли к помощи n спонсоров. У спонсоров есть требования к названиям задач. Название каждой задачи олимпиады — это некоторая строка, состоящая из m строчных букв латинского алфавита. i -й спонсор требует, чтобы в названии одной из задач на позициях l_i до r_i встречалась подстрока w_i . Иными словами, в наборе задач должна быть задача с названием s , где $s_{l_i} \dots s_{r_i} = w_i$.

Конечно, можно просто составить набор из n задач, чтобы угодить каждому из спонсоров, но составлять новые задачи сложно, а поэтому требуется составить минимальный набор из k названий задач так, чтобы удовлетворить требованиям всех спонсоров.

Формат входных данных

В первой строке содержится пара чисел n и m ($1 \leq n \leq 500$, $1 \leq m \leq 1000$).

В каждой из последующих n строк содержится по паре чисел l_i , r_i и строка w_i , состоящая из строчных букв латинского алфавита ($1 \leq l_i \leq r_i \leq m$, $|w_i| = r_i - l_i + 1$) — требование на название, которое предлагает i -й спонсор.

Формат выходных данных

В первой строке выведите число k ($1 \leq k \leq n$) — минимальное количество задач, которое нужно составить чернорусам.

В каждой из k последующих строк выведите названия задач — строки из строчных латинских букв длиной m .

Обратите внимание, что выведенные строки должны удовлетворять условию задачи, то есть для каждого спонсора должно найтись хотя бы одно название, которое удовлетворяет его требованиям.

Примеры

стандартный ввод	стандартный вывод
5 7 3 3 a 5 7 aba 3 5 aca 1 5 abaca 1 4 aaaa	2 abacaba aaaaaba
8 18 6 10 brest 11 17 vitebsk 1 5 gomel 6 11 grodno 1 5 minsk 12 18 mogilev 3 3 n 10 10 w	3 gomelbrestvitebska minskgrodnomogilev aaaaaaaaaaaaaaaa
5 9 1 9 aaaaaaaaaa 2 8 aaaaaaaa 3 7 aaaaaa 4 6 aaa 5 5 a	1 aaaaaaaaaa

Задача К. Регулярная задача

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Вам дан граф из n вершин и $2n$ ребер такой, что степень каждой вершины равна 4. Требуется выделить в нем не пересекающиеся по вершинам циклы суммарной длиной n .

Гарантируется, что для каждого набора входных данных это возможно.

Формат входных данных

В первой строке указано число n ($1 \leq n \leq 500\,000$).

В следующих $2n$ строках указаны пары чисел u и v ($1 \leq u, v \leq n, u \neq v$) — ребра графа. Гарантируется, что в графе нет кратных ребер.

Формат выходных данных

В первой строке укажите количество циклов.

В каждой последующей строке следует описать очередной цикл. Сначала указывается длина цикла, а затем его вершины, в порядке обхода.

Пример

стандартный ввод	стандартный вывод
6	2
1 2	3 1 2 3
1 3	3 4 5 6
1 5	
1 6	
2 3	
2 4	
2 5	
3 4	
3 6	
4 5	
4 6	
5 6	

Задача L. Гусеницы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

На числовой прямой выстроилось множество гусениц. Известно, что гусеницы разбиты на группы, при этом в i -й группе c_i гусениц, а находится эта группа в позиции x_i .

Есть группы гусениц, которые любят ходить к другим гусеницам в гости, а есть гусеницы, которые принимают гостей. Именно, тип групп гусениц задан параметром t , и если $t_i = 1$, то гусеницы из i -й группы ходят в гости (то есть каждая из этой группы), а если $t_i = 2$, то гусеницы из i -й группы принимают гостей.

У каждой гусеницы также есть уровень зрения, который совпадает для всех гусениц в одной группе и для группы i равен v_i . Если суммарный уровень зрения некоторых двух гусениц, одна из которых в группе i с $t_i = 1$, а другая – в группе j с $t_j = 2$, превосходит расстояние между этими гусеницами, то гусенице из группы i неинтересно идти к гусенице из группы j в гости, ведь они хотят за чашкой чая узнавать что-то новое о числовой прямой, и им интересно обсуждать те её участки, которые не видит ни одна из пары этих гусениц (если одна видит какой-то участок, а другая нет, то одна может позвонить другой по телефону и рассказать это). Наоборот, если суммарный уровень зрения этих гусениц не больше расстояния между ними, то этим двум гусеницам будет интересно обсудить пройденный гусеницей из группы i путь за чашкой чая.

Более формально, гусеница из группы i может пойти в гости к гусенице из группы j только в том случае, если $|x_j - x_i| \geq v_j + v_i$ и $t_i = 1$ и $t_j = 2$.

Гусеницы из одной ходящей в гости группы в итоге могут разойтись по разным группам, также как и гусеницы одной принимающей гостей группы могут принимать гусениц из разных групп. В итоге каждая гусеница из путешествующей группы хочет прийти к какой-то в гости, так как любит путешествовать и не может сидеть дома. При этом принимающие гости гусеницы любят гостей, но вполне себе смиряются, если к ним никто не пройдет. Итак, ваша задача – по имеющимся данным о координатах, типах, количествах и уровнях зрения групп гусениц определить, возможно ли распределить всех гусениц первого типа по гусеницам второго типа, чтобы выполнялось описанное выше ограничение.

Также обратите внимание, что может возникнуть ситуации, когда есть гусеницы только первого типа (тогда их невозможно распределить) или только второго типа (тогда ответ «Yes»).

Формат входных данных

В первой строке вам дано число $1 \leq t \leq 3 \cdot 10^5$, обозначающее количество тестовых примеров.

Следующие строки содержат описания тестовых примеров. Каждый тестовый пример содержит следующую информацию. В первой строке вам дано единственное число – количество групп гусениц $1 \leq n \leq 3 \cdot 10^5$. В следующей строке вам даны координаты групп гусениц в порядке возрастания – n целых чисел $-10^9 \leq x_1 < x_2 < x_3 < \dots < x_n \leq 10^9$. В следующей строке вам даны количества гусениц в порядке слева направо – n натуральных чисел $1 \leq c_i \leq 10^9$. В следующей строке вам даны n параметров $1 \leq v_i \leq 10^9$ – натуральные числа, обозначающие уровни зрения. В следующей строке вам даны типы групп гусениц слева направо – n натуральных чисел $1 \leq t_i \leq 2$. Гарантируется, что сумма n по всем тестовым наборам не превосходит $3 \cdot 10^5$.

Формат выходных данных

Для каждого теста в отдельной строке выведите «Yes» (без кавычек), если возможно всех путешествующих гусениц отправить в гости, и «No» (без кавычек) иначе.

Пример

стандартный ввод	стандартный вывод
4	Yes
2	No
10 20	No
1000000000 1000000000	Yes
4 4	
1 2	
2	
10 20	
1000000000 1000000000	
6 6	
1 2	
4	
0 9 13 15	
2 1 3 2	
6 6 1 1	
1 2 2 1	
4	
0 9 13 15	
2 1 3 2	
6 4 1 1	
1 2 2 1	