

Задача А. Ассистенты

Имя входного файла:	<code>standard input</code>
Имя выходного файла:	<code>standard output</code>
Ограничение по времени:	10 секунд
Ограничение по памяти:	512 мегабайт

Профессор Байтазар возглавляет крупный научный проект. До завершения проекта ему осталось провести n независимых экспериментов (пронумерованных целыми числами от 1 до n).

Каждый эксперимент профессор может провести самостоятельно или перепоручить ассистенту. Если Байтазар решает провести i -й эксперимент самостоятельно, он должен затратить p_i подряд идущих дней на него. При этом он не может заниматься другими делами. Если Байтазар перепоручит эксперимент ассистенту, то он должен потратить один день на подбор ассистента (так как Байтазар привык всё делать основательно, то в этот день он не может отвлекаться на другие дела). Найденный ассистент начинает работу на следующий день и проведёт эксперимент в течение a_i подряд идущих дней. При этом профессор может заниматься какими-то другими делами. Считается, что Байтазар всегда может подобрать квалифицированного ассистента, который ещё не задействован в проекте.

Профессор попросил Вас составить расписание его действий так, чтобы завершить проект как можно раньше.

Формат входного файла

Первая строка ввода содержит целое число n ($1 \leq n \leq 500\,000$) — количество экспериментов, оставшихся Байтазару до завершения проекта.

Далее следует n строк, i -я из них содержит два целых числа p_i и a_i ($1 \leq p_i \leq a_i \leq 10^9$), обозначающие количество дней, которые требуются для i -го эксперимента профессору и ассистенту соответственно.

Формат выходного файла

В первой строке выведите одно целое число k — минимальное количество дней, которое потребуется для завершения проекта. В последующих n строках выведите расписание экспериментов, позволяющее завершить проект за k дней.

Далее должны следовать n строк, i -я из них должна содержать график i -го эксперимента: сначала идёт буква «В» в случае, если профессор проводит i -й эксперимент самостоятельно, или «А» в случае, если он перепоручает эксперимент ассистенту, а затем через пробел идёт число t_i ($1 \leq t_i \leq k$) — день, в который профессор начнёт работу над экспериментом (тогда эксперимент завершится в день $t_i + p_i - 1$), или в который он будет искать нужного ассистента (который будет работать с $(t_i + 1)$ -го дня по $(t_i + a_i)$ -й).

В случае, если оптимальных расписаний несколько, выведите любое.

Примеры

standard input	standard output
5 2 6 2 6 2 5 2 2 1 2	7 A 1 B 3 A 2 A 5 B 7
1 1 1	1 B 1

Задача В. Раздел королевства

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Правивший много лет король Байтландии Байтазар Большое Гнездо задумал приучить своих наследников к государственным делам. Для этого он собирается разделить своё королевство на герцогства и распределить между своими сыновьями. Проблема заключается в том, что детей у Байтазара много... реально много.

Для того, чтобы не провоцировать споров о старшинстве, Байтазар решил, что герцогства будут иметь попарно различную площадь. При этом некоторую часть младших наследников при необходимости можно оставить и без герцогства.

Королевство можно представить как прямоугольник $n \times m$, разбитый на $n \cdot m$ единичных квадратов. Каждое герцогство должно быть составлено из целого числа единичных квадратов и должно быть связным по стороне (то есть между каждыми двумя единичными квадратами, входящими в герцогство, существует путь, полностью составленный из входящих в герцогство единичных квадратов, в котором каждый следующий квадрат имеет с предыдущим общую сторону). Каждый квадрат должен принадлежать ровно одному герцогству.

Король планирует разделить королевство так, чтобы как можно большее количество наследников получило свои герцогства. Помогите ему сделать это.

Формат входного файла

Единственная строка входа содержит два целых числа n и m ($1 \leq n, m \leq 1000$), обозначающие размеры прямоугольника.

Формат выходного файла

В первой строке выведите одно целое число k : наибольшее количество наследников, которые могут получить своё герцогство. Каждая из последующих n строк должна содержать по m символов — заглавных латинских букв из множества $\{A, \dots, Z\}$ — отображения герцогств. Отдельному герцогству соответствует связная часть, составленная из одинаковых букв и граничащая по сторонам только с частями, обозначенными другими буквами. Отметим, что различные части при этом вполне могут быть обозначены одной и той же буквой. При этом для любых двух частей количество составляющих эти части букв должно быть различно.

Гарантируется, что для любого соответствующего условиям задачи разбиения существует способ обозначить вышеописанным образом его части, используя не более чем 26 букв.

Примеры

standard input	standard output
4 5	5 ABCCE BBCCE DDDEE DDEEE
1 12	4 AABCCCCAAAAA
8 3	6 CCC CCC ACB ACB ABB ABB CCC ABB

Задача С. Неквадраты

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `2 секунды`
Ограничение по памяти: `512 мегабайт`

Задано целое положительное число n . Выясните, может ли оно быть представлено в виде произведения k целых положительных чисел, ни одно из которых *не* является квадратом целого числа.

Формат входного файла

Первая строка ввода содержит целое число t ($1 \leq t \leq 10$) — количество тестовых случаев. Каждая из последующих t строк содержит один тестовый случай, состоящий из двух целых чисел n ($1 \leq n \leq 1\,000\,000\,000$) и k ($2 \leq k \leq 50$).

Формат выходного файла

Для каждого тестового случая выведите в отдельной строке слово «YES» в том случае, если существует такой набор из k положительных целых чисел a_i , что $n = a_1 \cdot a_2 \cdot \dots \cdot a_k$ и ни одно из a_i не является квадратом целого числа, и слово «NO» в противном случае.

Примеры

standard input	standard output
4	NO
1 2	YES
6 2	NO
7 2	YES
8 3	

Задача D. Столбики монет

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `2 секунды`
Ограничение по памяти: `512 мегабайт`

Некоторое количество золотых и серебряных монет одного размера собрано на столе в n столбиков. Столбики могут иметь различную высоту.

За одно действие мы можем поменять местами две монеты, принадлежащие к различным столбикам, в том случае, если они находятся на одной высоте относительно стола.

Назовём столбик *одноцветным*, если он состоит из монет одного цвета (только серебряных или только золотых). Вычислите, какое наибольшее количество одноцветных столбиков можно собрать, сделав произвольное количество вышеописанных действий.

Формат входного файла

Первая строка входа содержит одно целое число n ($1 \leq n \leq 1\,000\,000$) — количество столбиков на столе.

Каждая из последующих n строк состоит из букв «G» и «S» и описывает один столбик. Буква «G» обозначает золотую монету, буква «S» — серебряную. Монеты перечислены снизу вверх — от поверхности стола к верхней монете в столбике.

Общее количество монет на столе не превышает 1 000 000.

Формат выходного файла

Выведите одно целое число — наибольшее количество одноцветных столбиков, которое может быть собрано из исходной позиции с помощью применения описанных в условии действий.

Примеры

standard input	standard output
5 GS SG SSG SGS SSSSG	4
3 GGG GG G	3

Задача Е. Государственные дороги

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

В средние века на территории Байтландии существовало несколько государств. При этом границы государств постоянно изменялись, в результате чего крупные города переходили из одного государства в другое. Сейчас историки пытаются выяснить конфигурацию государств в разное время.

Один из применяемых ими способов основывается на упоминании в хрониках статуса соединяющих города дорог. Если в какой-то исторический момент дорога имеет статус государственной, то историки считают, что соединяемые ею города в этот момент точно принадлежат одному и тому же государству. Однако для передвижения использовались и дороги местного значения, о которых хроники даже не упоминают. Поэтому утверждение о том, что любые два города в одном государстве соединены государственной дорогой, в общем случае неверно, равно как и утверждение, что между любыми двумя городами, принадлежащими одному государству, можно было проехать исключительно по государственным дорогам.

Для проверки этого подхода историкам нужна система, сохраняющая данные о статусе дорог в хронологическом порядке и отвечающая на запросы относительно того, возможно ли, что в текущий (с точки зрения внутренней хронологии системы) момент заданные города — и только они — образовывали единое государство.

Формат входного файла

Первая строка ввода содержит два целых числа n и q ($1 \leq n \leq 1\,000\,000$, $1 \leq q \leq 2\,000\,000$) — количество городов в средневековой Байтландии и количество событий (записей в хрониках и запросов). Города занумерованы последовательными целыми числами от 1 до n .

Далее идут события и запросы, перечисленные в хронологическом порядке (запрос относится к тому моменту времени, когда произошли все перечисленные до него события, но не произошло ни одного события, перечисленного после него). Каждая из q строк обозначает событие или запрос и имеет следующий формат:

- «1 u v » (событие первого типа) обозначает, что дорога между городами u и v получила статус государственной ($1 \leq u < v \leq n$);
- «2 m » (событие второго типа) обозначает, что дорога, которая получила статус государственной в результате m -го с начала хроник события первого типа, перестала быть таковой (m может принимать значения от 1 до количества событий первого типа, произошедших перед данным);
- «3 k u_1 u_2 ... u_k » — запрос: могло ли на момент, описываемый всеми уже обработанными событиями, существовать государство, список городов которого состоял из k городов с номерами $\{u_1, u_2, \dots, u_k\}$ ($1 \leq k \leq n$, $1 \leq u_1 < u_2 < \dots < u_k \leq n$).

На всех упоминаемых в хрониках дорогах движение является двусторонним, при этом любые два различных города соединены не более чем одной дорогой.

На момент инициализации базы (до первого запроса) ни одна дорога не имела статуса государственной. Одна и та же дорога может менять статус с государственной на обычную и наоборот сколько угодно раз. Гарантируется, что каждое m встретится в событиях второго типа не более одного раза. Сумма всех значений k не превосходит 2 000 000.

Формат выходного файла

В ответ на каждый запрос выведите в отдельной строке «YES» в случае, если данный список городов мог быть полным списком городов некоторого государства в соответствующий момент, и «NO» в противном случае.

Примеры

standard input	standard output
4 10	YES
3 3 1 3 4	NO
1 1 2	NO
3 3 1 3 4	YES
1 2 3	YES
3 2 1 3	
3 3 1 2 3	
1 3 4	
1 2 4	
2 1	
3 3 2 3 4	

Задача F. Stickers

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 7 секунд
Ограничение по памяти: 512 мегабайт

В магазине за углом продаётся n различных видов стикеров, i -й вид содержит d -буквенное слово s_i , состоящее из заглавных латинских букв. Все стикеры одного вида идентичны. Можно купить сколько угодно стикеров каждого вида. Требуется составить слово t из продающихся в магазине стикеров, при этом разрешается наклеивать стикер поверх уже существующих. Запрещается отрезать части стикеров, а также как-либо ещё модифицировать их.

Более формально, изначально у нас есть строка z , состоящая из m символов «#» (здесь m — длина строки t). За одно действие мы можем выбрать любую из строк s_i и целое число p между 0 и $m - d$, и для каждого $j \in \{1, \dots, d\}$ заменить $(p + j)$ -й символ слова z на j -й символ слова s_i (символы в строках пронумерованы с единицы).

Требуется найти минимальное количество таких действий, после которого получится требуемое слово t .

Формат входного файла

Первая строка ввода содержит два целых числа n и d ($1 \leq n, d \leq 50$) — количество видов стикеров и количество букв, написанных на каждом стикере.

i -я из последующих n строк задаёт один из видов стикеров и содержит ровно d заглавных латинских букв — слово s_i , написанное на стикере i -го вида.

Последняя строка ввода задаёт требуемое слово t , состоящее из не менее чем одной и не более чем 10^4 заглавных латинских букв.

Формат выходного файла

Выведите одно целое число — минимальное количество стикеров, которое потребуется для того, чтобы составить слово t . Если составить слово из данного набора стикеров невозможно, выведите слово «NO» вместо числа.

Примеры

standard input	standard output
2 3 ABA BCB ABACABA	3
2 4 ABBA BCBA ABBCBBA	NO