

Задача А. Пешки

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

В первом классе Глеб увлекался шахматами. К тому моменту он знал только лишь как ходит пешка: она может бить по диагонали влево-вверх и вправо-вверх, и ходить на клетку вверх только если та клетка не занята другой фигурой. О том, что пешка может превращаться в ферзя Глеб не подозревает. Поэтому он придумал свой вариант шахмат.

Игра идёт на доске с N строками и M столбцами ($1 \leq N \leq 100$, $1 \leq M \leq 100$) по следующим правилам. В нижней строке, имеющей номер 1, стоят P белых пешек, белых фигур на доске больше нет. На остальной части доски стоят разные чёрные фигуры (их названия Глеб не знает). Ходят только белые, их цель — побить все чёрные фигуры.

Как и в настоящих шахматах, если пешка Глеба бьёт чёрную фигуру, то она становится на её место, а побитая фигура убирается с доски. Считается, что Глеб выиграл, если он сумел побить белыми пешками все чёрные фигуры, в противном случае он проиграл. Помогите ему по заданной конфигурации всех фигур определить, сможет ли он выиграть, и, в случае успеха, выведите правильную последовательность ходов белых пешек.

Формат входных данных

Сначала вводятся четыре целых числа N , M , P , K ($1 \leq N \leq 100$, $1 \leq M \leq 100$, $0 \leq P \leq M$, $1 \leq K \leq 1000$, $K \leq (M - 1)N$). Далее записано P различных чисел — номера столбцов p_j ($1 \leq p_j \leq M$), в которых стоят белые пешки. Далее идут K различных пар целых чисел — координаты (строки и столбцы) чёрных фигур r_i , c_i ($2 \leq r_i \leq N$, $1 \leq c_i \leq M$).

Формат выходных данных

Если пешки не смогут съесть все фигуры, выведите единственное слово `NO`.

В противном случае в первую строку выведите `YES`, вторая строка должна содержать суммарное число перемещений C , последующие C строк — описание ходов пешек, по одному ходу на каждую строку. Каждый ход задаётся двумя координатами r , c пешки (номерами строки и столбца), которая будет ходить, и символом m , принимающем три значения: `L`, `R`, `F` — побить вперед и влево, побить вперед и вправо, сделать шаг вперед соответственно. Данные о ходе следует выводить разделёнными одним пробелом, сначала координаты, потом тип хода.

Если последовательностей ходов несколько, выведите любой из них. Обратите внимание, что минимизировать количество перемещений не требуется.

Примеры

<code>pawns.in</code>	<code>pawns.out</code>
2 2 2 1 1 2 2 2	YES 1 1 1 R
3 3 2 2 1 3 3 1 3 3	NO

Задача В. Декомпозиция потока

Имя входного файла: `decomposition.in`
Имя выходного файла: `decomposition.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером n и постройте декомпозицию этого потока.

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leq n \leq 500$, $1 \leq m \leq 10000$). Следующие m строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят 10^9 .

Формат выходных данных

В первую строку выходного файла выведите одно число — количество путей в декомпозиции максимального потока из вершины с номером 1 в вершину с номером n . Следующий строки должны содержать описания элементарных потоков, на который был разбит максимальный. Описание следует выводить в следующем формате: величина потока, количество ребер в пути, вдоль которого течет данный поток и номера ребер в этом пути. Ребра нумеруются с единицы в порядке появления во входном файле.

Примеры

<code>decomposition.in</code>	<code>decomposition.out</code>
4 5	3
1 2 1	1 2 1 4
1 3 2	1 3 2 3 4
3 2 1	1 2 2 5
2 4 2	
3 4 1	

Задача С. Охлаждение реактора

Имя входного файла: cooling.in
Имя выходного файла: cooling.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Известная террористическая группа под руководством знаменитого террориста Бен Гадена решила построить атомный реактор для получения оружейного плутония. Вам, как компьютерному гению этой группы, поручили разработать систему охлаждения реактора.

Система охлаждения реактора представляет собой набор труб, соединяющих узлы. По трубам течет жидкость, причем для каждой трубы строго определено направление, в котором она должна по ней течь. Узлы системы охлаждения занумерованы от 1 до N . Система охлаждения должна быть спроектирована таким образом, чтобы для каждого узла за единицу времени количество жидкости, втекающей в узел, было равно количеству жидкости, вытекающей из узла. То есть если из i -го узла в j -ый течет f_{ij} единиц жидкости за единицу времени (если из i в j нет трубы, то положим $f_{ij} = 0$), то для каждого узла i должно выполняться

$$\sum_{j=1}^N f_{ij} = \sum_{j=1}^N f_{ji}$$

У каждой трубы имеется пропускная способность c_{ij} . Кроме того, для обеспечения достаточного охлаждения требуется, чтобы по трубе протекало не менее l_{ij} единиц жидкости за единицу времени. То есть для трубы, ведущей из i -го узла в j -ый должно выполняться $l_{ij} \leq f_{ij} \leq c_{ij}$.

Вам дано описание системы охлаждения, выясните, каким образом можно пустить жидкость по трубам, чтобы выполнялись все указанные условия.

Формат входных данных

Первая строка входного файла содержит числа N и M – количество узлов и труб ($1 \leq N \leq 200$). Следующие M строк содержат описание труб. Каждая строка содержит четыре целых числа i, j, l_{ij} и c_{ij} . Любые два узла соединены не более чем одной трубой, если есть труба из i в j , то нет трубы из j в i , никакой узел не соединен трубой сам с собой, $0 \leq l_{ij} \leq c_{ij} \leq 10^5$.

Формат выходных данных

Если решение существует, выведите на первой строке выходного файла слово YES. Затем выведите M чисел – количество жидкости, которое должно течь по трубам, числа должны быть выведены в том порядке, в котором трубы заданы во входном файле. Если решения не существует, выведите NO.

Примеры

cooling.in	cooling.out
4 6 1 2 1 2 2 3 1 2 3 4 1 2 4 1 1 2 1 3 1 2 4 2 1 2	NO
4 6 1 2 1 3 2 3 1 3 3 4 1 3 4 1 1 3 1 3 1 3 4 2 1 3	YES 1 2 3 2 1 1

Задача D. Чаепитие

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

В одном из отделов крупной организации работает n человек. Как практически все сотрудники этой организации, они любят пить чай в перерывах между работой. При этом они достаточно дисциплинированы и делают в день ровно один перерыв, во время которого пьют чай. Для того, чтобы этот перерыв был максимально приятным, каждый из сотрудников этого отдела обязательно пьет чай одного из своих любимых сортов. В разные дни сотрудник может пить чай разных сортов. Для удобства пронумеруем сорта чая числами от 1 до m .

Недавно сотрудники отдела купили себе большой набор чайных пакетиков, который содержит a_1 пакетиков чая сорта номер 1, a_2 пакетиков чая сорта номер 2, ..., a_m пакетиков чая сорта номер m . Теперь они хотят знать, на какое максимальное число дней им может хватить купленного набора так, чтобы в каждый из дней каждому из сотрудников доставался пакетик чая одного из его любимых сортов.

Каждый сотрудник отдела пьет в день ровно одну чашку чая, которую заваривает из одного пакетика. При этом пакетики чая не завариваются повторно.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 50$). Вторая строка содержит m целых чисел a_1, \dots, a_m ($1 \leq a_i \leq 10^6$ для всех i от 1 до m).

Далее следуют n строк — i -я из этих строк описывает любимые сорта i -го сотрудника отдела и имеет следующий формат: сначала следует положительное число k_i — количество любимых сортов чая этого сотрудника, а затем идут k_i различных чисел от 1 до m — номера этих сортов.

Формат выходных данных

Выведите одно целое число — искомое максимальное количество дней.

Примеры

<code>tea.in</code>	<code>tea.out</code>
2 3 3 2 1 2 1 2 2 1 3	3

Задача Е. Карточки

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

Очередной день в ЛКШ. Сегодня вечером вы не захотели пойти на интеллектуалку. Вместо этого вы придумали себе более увлекательное занятие: перед вами разложено n карточек, с обеих сторон которых написаны буквы. Перед уходом ваш друг написал какую то строчку, оставив записку рядом. И вот вы придумали задачу — а возможно ли сложить эту строчку, используя имеющиеся карты? Каждую карту можно переворачивать и менять местами с другими. Более того, некоторые карты можно вовсе не использовать. Ваша задача — ответить на поставленный вопрос.

Формат входных данных

На первой строке стандартного ввода будет задано число n ($1 \leq n \leq 100\,000$) — количество карточек перед вами. В последующих n строках будут заданы сами карточки. Каждая из этих строк содержит ровно по две строчные латинские буквы. Первая из них — символ, написанный на лицевой стороне, а вторая — символ на обратной стороне. В последней строке задана сама строка из маленьких латинских букв. Она непуста и состоит не более, чем из 100 000 символов.

Формат выходных данных

Если возможно сложить заданную строку, выведите набор чисел, равный по размеру длине строки. При этом i -е число должно быть равно по модулю номеру карточки, которая будет соответствовать i -ому символу строки. Это число должны быть положительным, если карточка взята лицевой стороной, и отрицательно, если карточка взята обратной стороной.

Примеры

стандартный ввод	стандартный вывод
3 ab ab bc bca	-1 -3 2
3 ab ab bc bcc	IMPOSSIBLE