

А. Просто поток

5 секунд, 1024 мегабайта

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Входные данные

В первой строке записано натуральное число N — количество узлов в системе ($2 \leq N \leq 100$). Известно, что источник имеет номер 1, а сток номер N . Во второй строке записано натуральное M ($1 \leq M \leq 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i , где A_i, B_i — номера узлов, которые соединяет данная труба ($A_i \neq B_i$), а C_i ($0 \leq C_i \leq 10^4$) — наибольшая допустимая скорость течения воды через данную трубу.

Выходные данные

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3} .

входные данные

```
2
2
1 2 1
2 1 3
```

выходные данные

```
4
1
-3
```

В. Разрез

2 секунды, 1024 мегабайта

Найдите минимальный разрез между вершинами 1 и n в заданном неориентированном графе.

Входные данные

На первой строке входного файла содержится n ($2 \leq n \leq 100$) — число вершин в графе и m ($0 \leq m \leq 400$) — количество ребер. На следующих m строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

Выходные данные

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

| входные данные |
|--------------------------------|
| 3 3 1 2 3 1 3 5 3 2 7 |
| выходные данные |
| 2 8 1 2 |

С. Улиточки

2 секунды, 1024 мегабайта

Две улиточки Маша и Петя сейчас находятся в лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). По соображениям гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Входные данные

В первой строке файла записаны четыре целых числа — n , m , s и t (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика). В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние). Ограничения: $2 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $s \neq t$.

Выходные данные

Если существует решение, то выведите YES и на двух отдельных строчках сначала последовательность лужаек для Машеньки (дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

входные данные

```
3 3 1 3
1 2
1 3
2 3
```

выходные данные

```
YES
1 3
1 2 3
```

Дан оргграф, найти два непересекающихся по ребрам пути из s в t , вывести вершины найденных путей.

Д. Групповой турнир

2 секунды, 1024 мегабайта

В нашем капиталистическом и меркантильном мире всё решают деньги, и даже спорт не стал исключением. Все команды-участницы уже купили себе нужное количество очков в следующем сезоне, и местной федерации хоккея осталось только распределить результаты предстоящих игр. Однако, некоторые команды не поскупились и помимо покупки очков также купили ещё и результаты некоторых игр. Поначалу в федерации думали, что это им только упростит задачу: чем для большего числа игр результаты уже определены, тем меньше работы остаётся им. Но позже они поняли, что ошиблись. Они попросили вас стать участником их коррупционной схемы и помочь с распределением результатов игр предстоящего сезона.

Местный хоккейный турнир проходит по круговой системе: в турнире участвуют N команд и каждая команда играет с каждой ровно одну игру. За игру команды получают очки по следующим правилам:

- Если победителя удалось выявить в основное время матча, то ему достаётся 3 очка, а проигравшему — 0.
- Если основное время закончилось вничью и для выявления победителя понадобилось дополнительное время (овертайм), то победителю дают 2 очка, а проигравшему — 1 очко. Овертайм не ограничен во времени и длится до тех пор, пока одна из команд не забьёт гол.

По итогам турнира очки команды определяются как сумма её очков по всем сыгранным играм.

Входные данные

В первой строке входного файла содержится целое число N — количество участников турнира ($2 \leq N \leq 100$). Команды занумерованы числами от 1 до N .

Следующие N строк файла содержат по N символов и представляют собой турнирную таблицу на данный момент. Символ a_{ij} в строке i ($1 \leq i \leq N$) на позиции j ($1 \leq j \leq N$) означает результат игры команды номер i с командой номер j и может быть одним из:

- 'W' — означает, что команда i обыграла команду j в основное время матча;
- 'w' — команда i обыграла команду j в овертайме;
- 'L' — команда i проиграла команде j в овертайме;
- 'L' — команда i проиграла команде j в основное время матча;
- '.' — если результат игры между командами i и j ещё не определён;
- '#' — если i равно j , означает отсутствие данного матча, т. к. команда не может играть сама с собой.

Гарантируется, что данная таблица корректна. Более формально:

- $a_{ij} = \#$ для всех $i = j$;
- если $a_{ij} = \cdot$, то $a_{ji} = \cdot$;
- $a_{ij} = W$ тогда и только тогда, когда $a_{ji} = L$;
- $a_{ij} = w$ тогда и только тогда, когда $a_{ji} = L$.

Последняя строка входного файла содержит N целых чисел p_i — количество очков, которое требуется набрать i -й команде ($1 \leq i \leq N$).

Выходные данные

В выходной файл выведите полностью заполненную турнирную таблицу в формате, аналогичном формату входного файла.

Гарантируется, что решение существует. Если решений несколько, то можно вывести любое из них.

| входные данные |
|--|
| 4 #..W .#w. .l#. L..# 8 6 3 1 |
| выходные данные |
| #www l#ww Ll#w LLl# |

Е. Великая стена

2 секунды, 1024 мегабайта

У короля Людовика двое сыновей. Они ненавидят друг друга, и король боится, что после его смерти страна будет уничтожена страшными войнами. Поэтому Людовик решил разделить свою страну на две части, в каждой из которых будет властвовать один из его сыновей. Он посадил их на трон в города A и B , и хочет построить минимально возможное количество фрагментов стены таким образом, чтобы не существовало пути из города A в город B .

Страну, в которой властвует Людовик, можно упрощенно представить в виде прямоугольника $m \times n$. В некоторых клетках этого прямоугольника расположены горы, по остальным же можно свободно перемещаться. Кроме этого, ландшафт в некоторых клетках удобен для строительства стены, в остальных же строительство невозможно.

При поездках по стране можно перемещаться из клетки в соседнюю по стороне, только если ни одна из этих клеток не содержит горы или построенного фрагмента стены.

Входные данные

В первой строке входного файла содержатся числа m и n ($1 \leq m, n \leq 50$). Следующие m строк по n символов задают карту страны. Символы обозначают: «#» — гора, «.» — место, пригодное для постройки стены, «-» — место, не пригодное для постройки стены, «A» и «B» — города A и B .

Выходные данные

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены F , которые необходимо построить. Далее нужно вывести карту в том же формате, как во входном файле. Клетки со стеной обозначьте символом «+».

Если невозможно произвести требуемую застройку, то выведите в выходной файл единственное число - 1.

входные данные

```
5 5
--...
A-.#-
.#.#-
--.-
--.-B
```

выходные данные

```
3
--+..
A-+#-
+#.#-
--.-
--.-B
```

входные данные

```
1 2
AB
```

выходные данные

```
-1
```

входные данные

2 2
A#
#B

выходные данные

0
A#
#B