

## Задача А. Максимальный поток

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Найдите максимальный поток в заданной сети.

### Формат входных данных

В первой строке записаны два числа  $n$  и  $m$  ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 10\,000$ ) — число вершин и рёбер в сети, соответственно. Каждая из следующих  $m$  строк содержит по три числа  $u_i$ ,  $v_i$  и  $c_i$  ( $1 \leq u_i, v_i \leq n$ ,  $1 \leq c_i \leq 10\,000$ ), означающих, что между вершинами  $u_i$  и  $v_i$  в сети есть ребро с пропускной способностью  $c_i$ . Вершина 1 считается истоком, а вершина  $n$  — стоком. Граф сети считается неориентированным. Все числа во входных данных целые.

### Формат выходных данных

Выведите одно число — максимальный поток в заданной сети.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 2 3 1 3 5 3 2 7	8

## Задача В. Разрез

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

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

### Формат входных данных

В первой строке содержится  $n$  — число вершин в графе и  $m$  — количество рёбер ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 400$ ). В следующих  $m$  строках содержится описание рёбер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее десяти тысяч), при этом никакие две вершины не соединяются более чем одним ребром.

### Формат выходных данных

В первой строке должны содержаться количество рёбер в минимальном разрезе и их суммарная пропускная способность. В следующей строке выведите возрастающую последовательность номеров рёбер (рёбра нумеруются в том порядке, в каком они были заданы во входных данных).

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3	2 8
1 2 3	1 2
1 3 5	
3 2 7	

## Задача С. Разрезание графа

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Разбейте множество вершин заданного графа на два непустых подмножества  $A$  и  $B$  так, чтобы количество рёбер между вершинами различных подмножеств было минимально.

### Формат входных данных

В первой строке записано целое число  $n$  ( $2 \leq n \leq 50$ ) — число вершин в графе. Каждая из следующих  $n$  строк содержит по  $n$  символов.  $i$ -й символ  $j$ -й из этих строк равен «1», если между вершинами  $i$  и  $j$  есть ребро, и «0» в противном случае. Заданная таким образом матрица смежности графа является антирефлексивной (на главной диагонали стоят нули) и симметричной (относительно главной диагонали).

### Формат выходных данных

Выведите две строки. В первой выведите номера вершин, попавших во множество  $A$ , через пробел, а во второй — номера вершин, попавших во множество  $B$ , также через пробел. Номера вершин можно выводить в любом порядке.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	3
0111	1 2 4
1001	
1001	
1110	

## Задача D. Хакеры

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В сети компании Melksoft  $N$  серверов, использующих операционную систему Losedows. Некоторые из них соединены двусторонними каналами связи. Сеть называется надёжной, если между двумя любыми различными серверами найдётся какой-либо маршрут, состоящий из одного или нескольких каналов связи. Однако, сеть, в которой вообще нет серверов, надёжной не считается.

Наши доблестные хакеры хотят наглядно продемонстрировать компании Melksoft ошибку в последней версии Losedows (естественно, без согласия компании Melksoft). А именно, если в сети вырубить несколько серверов таким образом, что оставшаяся часть сети станет ненадёжной, то все оставшиеся серверы сети резко повисают.

Так как бомбардировка сервера битыми пакетами таким образом, чтобы он вырубился — занятие крайне тяжёлое, то хакеры хотят вырубить минимально возможное число серверов таким образом, чтобы все остальные повисли.

Напишите программу, которая определяет минимальное множество серверов, которые нужно бомбардировать.

### Формат входных данных

В первой строке заданы два числа  $N$  и  $M$  ( $1 \leq N \leq 50$ ,  $0 \leq M \leq 100$ ). Далее следуют  $M$  строк, описывающие пары серверов, соединённые каналами связи. Каждый канал описывается строкой из двух чисел  $u_i$   $v_i$ , где  $1 \leq u_i, v_i \leq N$  — номера серверов, соединённых  $i$ -м каналом. Два сервера могут быть соединены более чем одним каналом.

### Формат выходных данных

В первой строке выведите минимальное число серверов  $K$ , которое необходимо вырубить, чтобы все оставшиеся повисли из-за ошибки в Losedows. Во второй строке выведите номера серверов, которые необходимо вырубить, в произвольном порядке. Если оптимальных решений несколько, разрешается выводить любое. Если исходная сеть Melksoft ненадёжна, выведите число 0.

## Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 0	1 1
2 1 1 2	2 2 1
4 4 1 2 2 3 3 4 4 1	2 1 3
7 6 1 2 2 3 1 4 4 5 1 6 6 7	1 4

## Задача E. В поисках невест

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Однажды король Флатландии решил отправить  $k$  своих сыновей на поиски невест. Всем известно, что во Флатландии  $n$  городов, некоторые из которых соединены дорогами. Король живёт в столице, которая имеет номер 1, а город с номером  $n$  знаменит своими невестами.

Итак, король повелел, чтобы каждый из его сыновей добрался по дорогам из города 1 в город  $n$ . Поскольку, несмотря на обилие невест в городе  $n$ , красивых среди них не так много, сыновья опасаются друг друга. Поэтому они хотят добраться до цели таким образом, чтобы никакие два сына не проходили по одной и той же дороге (даже в разное время). Так как король любит своих сыновей, он хочет, чтобы среднее время сына в пути до города назначения было минимально.

Помогите придумать план передвижений, удовлетворяющий этим пожеланиям.

### Формат входных данных

В первой строке входных данных находятся числа  $n$ ,  $m$  и  $k$  — количество городов и дорог во Флатландии и количество сыновей короля, соответственно ( $2 \leq n \leq 200$ ,  $1 \leq m \leq 2000$ ,  $1 \leq k \leq 100$ ). Следующие  $m$  строк содержат по три целых положительных числа каждая — города, которые соединяет соответствующая дорога, и время, которое требуется для её прохождения (время не превышает  $10^6$ ). По дороге можно перемещаться в любом из двух направлений, два города могут быть соединены несколькими дорогами.

### Формат выходных данных

Если выполнить повеление короля невозможно, выведите в первой строке число  $-1$ . В противном случае выведите в первой строке минимальное возможное среднее время, которое требуется сыновьям, чтобы добраться до города назначения, не менее чем с пятью точными знаками после десятичной точки. В следующих  $k$  строках выведите пути сыновей: сначала число дорог в пути, а затем номера дорог в пути в том порядке, в котором их следует проходить. Дороги нумеруются начиная с единицы в том порядке, в котором они заданы во входных данных.

## Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 8 2	3.00000
1 2 1	3 1 5 6
1 3 1	3 2 7 8
1 4 3	
2 5 5	
2 3 1	
3 5 1	
3 4 1	
5 4 1	

## Задача F. План эвакуации

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В городе есть муниципальные здания и бомбоубежища, которые были специально построены для эвакуации служащих в случае ядерной войны. Каждое бомбоубежище имеет ограниченную вместительность по количеству людей, которые могут в нём находиться. В идеале все работники из одного муниципального здания должны были бы бежать к ближайшему бомбоубежищу. Однако, в таком случае, некоторые бомбоубежища могли бы переполниться, в то время как остальные остались бы наполовину пустыми.

Чтобы разрешить эту проблему, Городской Совет разработал специальный план эвакуации. Вместо того, чтобы каждому служащему индивидуально приписать, в какое бомбоубежище он должен бежать, для каждого муниципального здания определили, сколько служащих из него в какое бомбоубежище должны бежать. Задача индивидуального распределения была переложена на внутреннее управление муниципальных зданий.

План эвакуации учитывает количество служащих в каждом здании — каждый служащий должен быть учтён в плане, и в каждое бомбоубежище может быть направлено количество служащих, не превосходящее вместимости бомбоубежища.

Городской Совет заявляет, что их план эвакуации оптимален в том смысле, что суммарное время эвакуации всех служащих города минимально.

Мэр города, находящийся в постоянной конфронтации с Городским Советом, не слишком то верит этому заявлению. Поэтому он нанял Вас в качестве независимого эксперта для проверки плана эвакуации. Ваша задача состоит в том, чтобы либо убедиться в оптимальности плана Городского Совета, либо доказать обратное, представив в качестве доказательства другой план эвакуации с меньшим суммарным временем для эвакуации всех служащих.

Карта города может быть представлена в виде квадратной сетки. Расположение муниципальных зданий и бомбоубежищ задается парой целых чисел, а время эвакуации из муниципального здания с координатами  $(X_i, Y_i)$  в бомбоубежище с координатами  $(P_j, Q_j)$  составляет  $D_{ij} = |X_i - P_j| + |Y_i - Q_j| + 1$  минут.

## Формат входных данных

Входные данные содержат описание карты города и плана эвакуации, предложенного Городским Советом. Первая строка входных данных содержит два целых числа  $N$  ( $1 \leq N \leq 100$ ) и  $M$  ( $1 \leq M \leq 100$ ), разделенных пробелом.  $N$  — число муниципальных зданий в городе (все они занумерованы числами от 1 до  $N$ ),  $M$  — число бомбоубежищ (все они занумерованы числами от 1 до  $M$ ).

Последующие  $N$  строк содержат описания муниципальных зданий. Каждая строка содержит целые числа  $X_i, Y_i$  и  $B_i$ , разделенные пробелами, где  $X_i, Y_i$  ( $-1000 \leq X_i, Y_i \leq 1000$ ) — координаты здания, а  $B_i$  ( $1 \leq B_i \leq 1000$ ) — число служащих в здании.

Описание бомбоубежищ содержится в последующих  $M$  строках. Каждая строка содержит целые числа  $P_j, Q_j$  и  $C_j$ , разделенные пробелами, где  $P_j, Q_j$  ( $-1000 \leq P_j, Q_j \leq 1000$ ) — координаты бомбоубежища, а  $C_j$  ( $1 \leq C_j \leq 1000$ ) — вместимость бомбоубежища.

В последующих  $N$  строках содержится описание плана эвакуации. Каждая строка представляет собой описание плана эвакуации для отдельного здания. План эвакуации из  $i$ -го здания состоит из  $M$  целых чисел  $E_{ij}$ , разделенных пробелами.  $E_{ij}$  ( $0 \leq E_{ij} \leq 10000$ ) — количество служащих, которые должны эвакуироваться из  $i$ -го здания в  $j$ -е бомбоубежище.

Гарантируется, что заданный план корректен.

## Формат выходных данных

Если план эвакуации Городского Совета оптимален, то выведите одно слово OPTIMAL. В противном случае выведите на первой строке слово SUBOPTIMAL, а в последующих  $N$  строках выведите Ваш план эвакуации (более оптимальный) в том же формате, что и во входных данных. Ваш план не обязан быть оптимальным, но должен быть лучше плана Городского Совета.

**Примеры**

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 4 -3 3 5 -2 2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 1 1 0 0 0 6 0 0 3 0 2	SUBOPTIMAL 3 0 1 1 0 0 6 0 0 4 0 1
3 4 -3 3 5 -2 2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 0 1 1 0 0 6 0 0 4 0 1	OPTIMAL

**Задача G. Максимальное паросочетание с весами** Примеры  
**рёбер**

Имя входного файла: *стандартный ввод*  
 Имя выходного файла: *стандартный вывод*  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 512 мегабайт

Дан двудольный граф. У каждой вершины графа есть вес. Вес ребра — сумма весов его концов. Вес паросочетания — сумма весов рёбер, входящих в паросочетание. Нужно найти паросочетание максимального веса. Заметим, это паросочетание может содержать сколько угодно рёбер, единственное условие — вес паросочетания должен быть максимальным.

Напомним, что паросочетанием в двудольном графе называется набор рёбер этого графа такой, что никакие два ребра набора не имеют общих вершин.

**Формат входных данных**

В первой строке заданы размеры долей  $n$  и  $m$  ( $1 \leq n, m \leq 5000$ ) и количество рёбер  $e$  ( $0 \leq e \leq 10\,000$ ). Вторая строка содержит  $n$  целых чисел от 0 до 10 000 — веса вершин первой доли. Третья строка содержит  $m$  целых чисел от 0 до 10 000 — веса вершин второй доли. Следующие  $e$  строк содержат рёбра графа. Каждое ребро описывается парой целых чисел  $a_i b_i$ , где  $1 \leq a_i \leq n$  — номер вершины первой доли и  $1 \leq b_i \leq m$  — номер вершины второй доли.

**Формат выходных данных**

В первой строке выведите  $w$  — максимальный вес паросочетания. Во второй строке выведите  $k$  — количество рёбер в паросочетании максимального веса. В следующей строке выведите  $k$  различных чисел от 1 до  $e$  — номера рёбер в паросочетании. Если максимальных по весу паросочетаний несколько, разрешается вывести одно любое.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 3 3 2 0 9 9 1 0 9 1 2 2 1 1 1	3 1 3
3 2 4 1 2 3 1 2 1 1 2 1 2 2 3 2	8 2 4 2

## Задача Н. Пути

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Найдите размер максимального множества вершинно-непересекающихся путей между вершинами 1 и 2 в заданном неориентированном графе.

### Формат входных данных

В первой строке задано число вершин  $N$  ( $2 \leq N \leq 400$ ) и ребер  $M$  ( $1 \leq M \leq 10\,000$ ). Далее следуют  $M$  строк, описывающих начало и конец соответствующего ребра.

### Формат выходных данных

Выведите искомое число.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
7 10 1 3 1 5 1 6 2 4 2 5 2 7 3 4 3 5 5 7 6 7	3

## Задача I. Непересекающиеся пути — лайт

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из  $r$  строк и  $c$  столбцов. Каждая клетка либо свободна, либо занята стеной.

Несколько улиток начинают движение в свободной клетке, обозначенной на карте буквой «R». За один шаг улитка может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выползть за пределы лабиринта нельзя.

Задача улиток — оказаться в свободной клетке, обозначенной на карте буквой «G». Но при этом нужно, чтобы каждую клетку лабиринта либо вообще не посетили улитки, либо один раз посетила одна из улиток. Исключение составляют начальная и конечная клетки: их может по одному разу посетить каждая улитка, а не только какая-то одна.

Сколько улиток могут по этим правилам добраться до конечной клетки? Найдите пути, позволяющие как можно большему количеству улиток это сделать. Количество шагов в найденных путях может быть любым.

Гарантируется, что начальная и конечная клетки не являются соседними.

### Формат входных данных

В первой строке записаны два целых числа  $r$  и  $c$  — количество строк и столбцов в карте лабиринта ( $1 \leq r, c \leq 50$ ). Каждая из следующих  $r$  строк содержит  $c$  символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу и не находятся в соседних клетках.

### Формат выходных данных

В первой строке выведите целое число  $k$  — количество найденных путей, оно должно быть максимальным возможным. В следующих  $k$  строках выведите сами пути. Каждый путь должен быть выведен как последовательность шагов, начинаться в начальной клетке и заканчиваться в конечной. Все остальные клетки всех путей должны быть попарно различны и отличаться от начальной и конечной клеток. Количество шагов в выведенных путях может быть любым.

Обозначения шагов для движения по карте лабиринта:

- «N» (север) — вверх,
- «W» (запад) — влево,
- «S» (юг) — вниз,
- «E» (восток) — вправо.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 # . . . . . . # . R . G . . .	2 NWWW SWSE WSWW
2 2 R# #G	0

## Задача J. Непересекающиеся пути

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из  $r$  строк и  $c$  столбцов. Каждая клетка либо свободна, либо занята стеной.

Несколько улиток начинают движение в свободной клетке, обозначенной на карте буквой «R». За один шаг улитка может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выползть за пределы лабиринта нельзя.

Задача улиток — оказаться в свободной клетке, обозначенной на карте буквой «G». Но при этом нужно, чтобы каждую клетку лабиринта либо вообще не посетили улитки, либо один раз посетила одна из улиток. Исключение составляют начальная и конечная клетки: их может по одному разу посетить каждая улитка, а не только какая-то одна.

Сколько улиток могут по этим правилам добраться до конечной клетки? Найдите пути, позволяющие как можно большему количеству улиток это сделать. При этом суммарное количество шагов в найденных путях должно быть наименьшим из возможных для такого количества улиток.

Гарантируется, что начальная и конечная клетки не являются соседними.

### Формат входных данных

В первой строке записаны два целых числа  $r$  и  $c$  — количество строк и столбцов в карте лабиринта ( $1 \leq r, c \leq 50$ ). Каждая из следующих  $r$  строк содержит  $c$  символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу и не находятся в соседних клетках.

### Формат выходных данных

В первой строке выведите целое число  $k$  — количество найденных путей, оно должно быть максимальным возможным. В следующих  $k$  строках выведите сами пути. Каждый путь должен быть выведен как последовательность шагов, начинаться в начальной клетке и заканчиваться в конечной. Все остальные клетки всех путей должны быть попарно различны и отличаться от начальной и конечной клеток. Суммарное количество шагов в выведенных путях должно быть наименьшим из возможных для  $k$  улиток.

Обозначения шагов для движения по карте лабиринта:

- «N» (север) — вверх,
- «W» (запад) — влево,
- «S» (юг) — вниз,
- «E» (восток) — вправо.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 # . . . . . . # . R . G . . .	2 NWWWSS WSWW
2 2 R# #G	0