

## Задача А. Паросочетание Юнга максимального веса

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

Дана диаграмма Юнга разложения числа  $2n$  в сумму двух одинаковых слагаемых. В каждой ячейке левого столбца диаграммы написано число, в  $i$ -й строке —  $w_i$ .

Некоторые ячейки левого столбца магически связаны с некоторыми ячейками правого столбца. Определим паросочетание Юнга (или юнговое паросочетание) как множество связей между ячейками, такое что каждая ячейка принадлежит не более чем одной связи. Вес юнгового паросочетания, в котором лежат ячейки левого столбца из строк  $a_1, a_2, \dots, a_k$  есть  $\sqrt{\sum_{i=1}^k w_{a_i}^2}$ . Требуется найти юнговое паросочетание максимального веса.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 1000$ ). Вторая строка входного файла содержит  $n$  целых чисел  $w_1, w_2, \dots, w_n$  ( $1 \leq w_i \leq 1000$ ). Следующие  $n$  строк содержат описания связей, соответствующих вершинам левого столбца. Формат описания: количество связей, затем номера строк правого столбца, разделенные пробелом. Суммарное количество связей не превосходит 200 000.

### Формат выходных данных

Выведите  $n$  чисел — для каждой ячейки левого столбца выведите строку ячейки правого столбца, с которой её надо взять в Юнговое паросочетание. Если вершина не входит в паросочетание, выведите 0.

### Примеры

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

## Задача В. Испорченный паркет

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

Пол в некоторой комнате размером  $M \times N$  замощен паркетом. При этом некоторые плитки паркета оказались испорчены. Петя решил сделать ремонт в этой комнате, заменив только испорченные клетки. Придя в магазин, он обнаружил, что паркетные плитки бывают двух типов — размера  $1 \times 2$ , которые стоят  $A$  рублей (немного подумав, Петя понял, что плитки  $1 \times 2$  можно поворачивать на 90 градусов, получая тем самым плитки  $2 \times 1$ ) и размера  $1 \times 1$ , которые стоят  $B$  рублей. Разрезать плитку размера  $1 \times 2$  на две размера  $1 \times 1$  Петя не может.

Определите, какая минимальная сумма денег нужна Пете, чтобы сделать ремонт.

### Формат входных данных

Первая строка входного файла содержит 4 числа  $N$ ,  $M$ ,  $A$ ,  $B$  ( $1 \leq N, M \leq 300$ ,  $A, B$  — целые числа, по модулю не превосходящие 1000). Каждая из последующих  $N$  строк содержит по  $M$  символов: символ «.» (точка) обозначает неиспорченную плитку паркета, а символ «\*» (звездочка) — испорченную. В конце строк могут идти незначащие пробелы. В конце файла могут быть пустые строки.

### Формат выходных данных

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно заменить испорченные паркетины (и только их).

### Примеры

floor.in	floor.out
2 3 3 2 .** .*.	5

## Задача С. 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

minimal.in	minimal.out
3 2	2
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

## Задача D. Такси

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунда
Ограничение по памяти:	256 мегабайт

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел:  $x$ -координатой и  $y$ -координатой. Время, необходимое для того, чтобы добраться из точки с адресом  $(a, b)$  в точку  $(c, d)$ , равно  $|a - c| + |b - d|$  минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

### Формат входных данных

В первой строке входного файла записано число заказов  $M$  ( $0 < M < 500$ ). Последующие  $M$  строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате `hh:mm` (в интервале с `00:00` по `23:59`), координаты  $(a, b)$  точки отправления и координаты  $(c, d)$  точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

### Формат выходных данных

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

### Примеры

taxi.in	taxi.out
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2

## Задача Е. Толстые хоббиты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Ни один хоббит не в состоянии в одиночку противостоять полчищам Мордора. . . В последний поход против Мордора Гэндальф решил отправить  $N$  хоббитов из Шира. Но часть хоббитов наотрез отказалась, жалуюсь на то, что другие хоббиты наверняка будут дразнить их толстыми. После опроса всех хоббитов оказалось, что любой хоббит отказывается принять участие в походе в том случае, если с ним в поход выступит хотя бы один хоббит с меньшим весом. К счастью для Средиземья, не все хоббиты знают свой точный вес. В Шире были всего одни весы чашечного типа, позволяющие для пары хоббитов определить, какой хоббит тяжелее. Некоторые пары хоббитов взвешивались на этих весах. Всем хоббитам известен результат всех взвешиваний. Гэндальф абсолютно уверен, что в Шире нет двух хоббитов одного веса. Он заинтересован в том, чтобы отряд состоял из наибольшего количества хоббитов. Однако найти наибольшее множество хоббитов, среди которых ни один не считает себя тяжелее другого, оказалось не так-то просто. Подскажите Гэндальфу, на сколько хоббитов он может рассчитывать. Помните при этом, что хоббиты умные существа и знают, что если Сэм тяжелее Пиппина, а Пиппин тяжелее Фродо, то Сэм и подавно будет тяжелее Фродо.

### Формат входных данных

В первой строке дано целое число  $N$  – количество хоббитов ( $2 \leq N \leq 100$ ). Все хоббиты пронумерованы целыми числами от 1 до  $N$ . В следующих  $N$  строках записана матрица размера  $N \times N$ . Если  $i$ -й и  $j$ -й хоббит взвешивались на чашечных весах и оказалось, что  $i$ -й хоббит тяжелее, то в  $i$ -й строке матрицы на  $j$ -й позиции стоит единица. Во всех остальных случаях в матрице стоят нули.

### Формат выходных данных

В первой строке выведите размер наибольшего множества хоббитов, готового выступить в поход, во второй строке перечислите номера хоббитов из этого множества через пробел.

### Примеры

hobbits.in	hobbits.out
2 0 1 0 0	1 2
3 0 0 0 0 0 0 0 0 0	3 1 2 3

## Задача F. Максимальное паросочетание

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 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, что 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  $e$  ( $0 \leq e \leq 10000$ ). 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.

### Примеры

стандартный ввод	стандартный вывод
4 3 3 2 0 9 9 1 0 9 1 2 2 1 1 1	3 2 1 2
3 2 4 1 2 3 1 2 1 1 2 1 2 2 3 2	8 2 2 4