

## Задача А. Яблоко от яблони

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

У Пети в саду растет яблоня. Воодушевленный историей об Исааке Ньютоне, который, как известно, открыл закон всемирного тяготения после того, как ему на голову упало яблоко, Петя с целью повысить свою успеваемость по физике часто сидит под яблоней.

Однако, поскольку по физике у Пети твердая тройка, яблоки с его яблони падают следующим образом. В какой-то момент одно из яблок отрывается от ветки, на которой оно висит, и начинает падать строго вниз. Если в некоторый момент оно задевает другое яблоко, то то тоже отрывается от своей ветки и начинает падать вниз, при этом первое яблоко не меняет направление своего падения. Вообще, если любое падающее яблоко заденет другое яблоко на своем пути, то оно также начнет падать.

Таким образом, в любой момент каждое яблоко либо висит на ветке, либо падает строго вниз, причем все яблоки кроме первого, чтобы начать падать, должны сначала соприкоснуться с каким-либо другим падающим яблоком.

Выясните, какие яблоки упадут с Петиной яблони.

### Формат входных данных

В первой строке вводится число  $N$  — количество яблок на Петиной яблоне ( $1 \leq N \leq 200$ ). Следующие  $N$  строк содержат описания яблок. Будем считать все яблоки шарами. Каждое яблоко задается координатами своей самой верхней точки (той, где оно исходно прикреплено к дереву, длиной черенка пренебрежем)  $x_i, y_i$  и  $z_i$  и радиусом  $r_i$  ( $-10000 \leq x_i, y_i, z_i \leq 10000, 1 \leq r_i \leq 10000$ , все числа целые). Гарантируется, что изначально никакие яблоки не пересекаются (даже не соприкасаются). Ось  $OZ$  направлена вверх.

**Вершины в ответе должны быть отсортированы по возрастанию.**

### Формат выходных данных

В первой строке выведите количество яблок, которые упадут с яблони, если начнет падать первое яблоко. В следующей строке выводите номера упавших яблок. Яблоки нумеруются, начиная с 1, в том порядке, в котором они заданы во входных данных.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4                | 3                 |
| 0 0 10 4         | 1 2 4             |
| 5 0 3 1          |                   |
| -7 4 7 1         |                   |
| 0 1 2 6          |                   |

## Задача В. Предок

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

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 100000$ ) — количество вершин в дереве. Во второй строке находятся  $n$  чисел,  $i$ -е из которых определяет номер непосредственного родителя вершины с номером  $i$ . Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число  $m$  ( $1 \leq m \leq 100000$ ) — количество запросов. Каждая из следующих  $m$  строк содержит два различных числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ).

### Формат выходных данных

Для каждого из  $m$  запросов выведите на отдельной строке число 1, если вершина  $a$  является одним из предков вершины  $b$ , и 0 в противном случае.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6                | 0                 |
| 0 1 1 2 3 3      | 1                 |
| 5                | 1                 |
| 4 1              | 0                 |
| 1 4              | 0                 |
| 3 6              |                   |
| 2 6              |                   |
| 6 5              |                   |

## Задача С. Введите одностороннее движение

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

В Тридевятом Царстве было  $N$  городов, некоторые из которых были соединены дорогами. К сожалению, в последнее время добраться из одного города в другой стало очень сложно из-за возникших автомобильных пробок. В целях борьбы с пробками было решено все дороги сделать односторонними, т.е. разрешить проезд по каждой дороге только в одном направлении. При этом требуется, чтобы по-прежнему можно было из любого города попасть в любой другой.

### Формат входных данных

Во входном файле записано сначала число  $N$  — количество городов ( $1 \leq N \leq 1000$ ). Затем записано число  $M$  — количество дорог ( $1 \leq M \leq 100000$ ). Далее идет  $M$  пар чисел, задающих дороги (каждая дорога описывается номерами городов, которые она соединяет). Не бывает дорог из некоторого города в тот же город. Между двумя городами может быть несколько дорог. Гарантируется, что до введения одностороннего движения можно было попасть из любого города в любой другой.

### Формат выходных данных

В выходной файл нужно выдать  $M$  пар чисел, соответствующих дорогам (дороги должны быть выданы в том же порядке, в котором они заданы во входном файле). Для каждой дороги сначала должен быть записан номер города, из которого по ней можно будет уехать после введения одностороннего движения, а затем — номер города, куда эта дорога ведет.

Если ввести одностороннее движение так, чтобы можно было из любого города попасть в любой другой, нельзя, выходной файл должен содержать одно число 0.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4                | 1 2               |
| 6                | 2 1               |
| 1 2              | 2 3               |
| 1 2              | 4 2               |
| 2 3              | 3 4               |
| 2 4              | 4 1               |
| 4 3              |                   |
| 1 4              |                   |

## Задача D. Авиалинии

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

В Берляндии скоро появятся свои авиалинии. Комитет по разработке берляндских авиалиний уже предложил свой вариант соединения городов авиарейсами. Каждый авиарейс задается парой различных городов. Рейсы односторонние. Президенту понравился план, однако он показался ему чересчур неэкономным. Требуется разработать новый план, который содержит наименьшее количество авиарейсов и удовлетворяет условию: если из города  $a$  можно было попасть в город  $b$  (возможно, с пересадками) согласно первоначальному плану, то и в новом плане это должно быть возможным. Если же это было сделать невозможно, то и согласно новому плану это не должно быть возможным. Очевидно, что из любого города можно попасть в него самого.

### Формат входных данных

В первой строке входного файла записаны целые числа  $N$  и  $M$  ( $1 \leq N \leq 10^3$ ,  $0 \leq M \leq 10^4$ ), где  $N$  — количество городов в стране, а  $M$  — количество авиарейсов в первоначальном плане. Города нумеруются от 1 до  $N$ . Далее записано  $M$  пар различных чисел  $a_i, b_i$  обозначающих наличие рейса из  $a_i$  в  $b_i$  в первоначальном плане ( $1 \leq a_i \leq N$ ,  $1 \leq b_i \leq N$ ). Пары разделяются пробелами или переводами строк. Между парой городов может быть более одного авиарейса.

### Формат выходных данных

В первую строку выходного файла выведите количество рейсов в новом плане. Далее выведите авиарейсы в формате, аналогичном формату входных данных. Пары разделяйте пробелами или переводами строк. Пары выводите в любом порядке. Если существует несколько решений, выведите любое.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 5              | 4                 |
| 1 2              | 1 2 2 3 3 1 1 4   |
| 2 3              |                   |
| 2 1              |                   |
| 3 2              |                   |
| 2 4              |                   |

## Задача Е. Монополия

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

В Тридесятом государстве есть  $N$  фирм, занимающихся разработкой программного обеспечения. Однажды известный олигарх Тридесятого государства Иванушка решил монополизировать эту отрасль. Для этого он хочет купить максимальное число программистских фирм Тридесятого государства.

Он разослал предложения всем  $N$  компаниям и через некоторое время получил от каждой их них согласие или отказ. Однако он знает, что в бизнесе очень многое зависит от взаимного доверия партнеров.

В результате небольшого исследования Иванушка установил, между какими компаниями существует взаимное доверие (причем всегда если компания доверяет компании  $B$ , то компания  $B$  доверяет компании  $A$ ).

Теперь, при желании, Иванушка может повторно разослать предложения всем компаниям, включив в письма список компаний, давших согласие участвовать в его проекте. При этом каждая компания, независимо от своего первоначального мнения дает согласие, если в списке есть хотя бы одна компания, которой она доверяет, и отказ в противном случае. Таким образом, некоторые компании, которые изначально не согласились участвовать в проекте, могут теперь дать свое согласие, а некоторые из давших согласие — наоборот отказаться. В результате этого у Иванушки формируется новый список, который он опять может разослать фирмам. Он может сколь угодно долго повторять операцию, каждый раз рассылая текущий список. Иванушка может остановить процесс в любой момент и заключить договора с теми, кто после последней рассылки дал согласие.

Напишите программу, которая определит, какое максимальное число компаний может объединить Иванушка под своим началом.

Будем считать, что Иванушка — честный предприниматель и он никогда не подтасовывает рассылаемые им списки.

### Формат входных данных

В первой строке входных данных содержится число  $N$  — количество фирм ( $1 \leq N \leq 2000$ ). Далее идут  $N$  чисел, описывающих ответ фирмы на первое предложение Иванушки (1 — согласие, 0 — отказ). Далее задается число  $M$  ( $0 \leq M \leq 200000$ ) — количество пар компаний, между которыми существует доверие. Далее следуют  $M$  пар чисел, задающих номера фирм, между которыми существует взаимное доверие (числа в паре не могут быть одинаковыми). Любая пара компаний упоминается в этом списке не более одного раза.

### Формат выходных данных

Выведите одно число — максимальное число фирм, которое сможет купить Иванушка.

### Пример

| стандартный ввод  | стандартный вывод |
|---|-------------------|
| 7<br>1 0 0 0 0 0 1<br>6<br>1 2<br>1 3<br>1 4<br>4 5<br>5 6<br>2 5 | 4                 |

## Задача F. Водостоки

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

Карту местности условно разбили на квадраты, и посчитали среднюю высоту над уровнем моря для каждого квадрата.

Когда идет дождь, вода равномерно выпадает на все квадраты. Если один из четырех соседних с данным квадратом квадратов имеет меньшую высоту над уровнем моря, то вода с текущего квадрата стекает туда (и, если есть возможность, то дальше), если же все соседние квадраты имеют большую высоту, то вода скапливается в этом квадрате.

Разрешается в некоторых квадратах построить водостоки. Когда на каком-то квадрате строят водосток, то вся вода, которая раньше скапливалась в этом квадрате, будет утекать в водосток.

Если есть группа квадратов, имеющих одинаковую высоту и образующих связную область, то если хотя бы рядом с одним из этих квадратов есть квадрат, имеющий меньшую высоту, то вся вода утекает туда, если же такого квадрата нет, то вода стоит во всех этих квадратах. При этом достаточно построить водосток в любом из этих квадратов, и вся вода с них будет утекать в этот водосток.

Требуется определить, какое минимальное количество водостоков нужно построить, чтобы после дождя вся вода утекала в водостоки.

### Формат входных данных

Во входном файле записаны сначала числа  $N$  и  $M$ , задающие размеры карты — натуральные числа, не превышающие 100. Далее идет  $N$  строк, по  $M$  чисел в каждой, задающих высоту квадратов карты над уровнем моря. Высота задается натуральным числом, не превышающим 10000. Считается, что квадраты, расположенные за пределами карты, имеют высоту 10001 (то есть вода никогда не утекает за пределы карты).

### Формат выходных данных

В выходной файл выведите минимальное количество водостоков, которое необходимо построить.

### Пример

| стандартный ввод                                | стандартный вывод |
|---|-------------------|
| 4 4<br>1 2 4 1<br>2 4 4 4<br>1 4 3 2<br>1 2 3 2 | 4                 |