

Russian Code Cup 2014
Квалификация 4

"А" Соцопрос

Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Когда члены жюри Russian Code Cup узнали о необходимости проведения четвертого квалификационного раунда, они решили провести специальное исследование, чтобы придумать для этого раунда самую простую задачу.

Социологическому отделу было поставлена задача провести исследование. Для некоторого типа задач требовалось определить, сколько человек смогут решить подобную задачу, и сколько человек будут считать задачу такого типа настолько отвратительной, что даже не захотят думать о ней.

Социологи восприняли задачу буквально и выдали следующий результат исследования. В раунде примут участие n человек. Из них a человек умеет решать задачи такого типа, и b человек считают такие задачи отвратительными и не будут пытаться их решать.

Теперь члены жюри пытаются понять, сколько человек в итоге решит задачу такого типа. Будем считать человек решит задачу, если он умеет решать такие задачи и не считает их отвратительными. К сожалению, по данным, полученным от социологов, узнать точное число участников, которые решат задачу, невозможно. Помогите выяснить, каким может оказаться минимальное и максимальное число людей, которое решит задачу на туре, если данные социологов верны.

Например, пусть в туре планирует участвовать 4 человека, 3 из умеют решать задачи некоторого типа, и 3 считают такие задачи отвратительными и не будут пытаться их решать. Если это одни и те же три человека, то никто не решит такую задачу, поэтому минимальное число равно 0. Если же один из тех, кто умеет решать такие задачи, не считает их отвратительными, то он решит такую задачу, поэтому максимальное число равно 1.

Формат входных данных

Первая строка содержит целое положительное число T — число тестовых примеров во входных данных. Гарантируется, что T не превышает 10000. Далее следуют описания тестовых примеров. Каждый тестовый пример описывается одной строкой. В этой строке содержится три целых числа: n , a , b — количество

Russian Code Cup 2014
Квалификация 4

участников, количество тех, кто умеет решать задачу некоторого типа, и количество тех, которые считают задачи этого типа отвратительными. ($1 \leq n \leq 1000$, $0 \leq a, b \leq n$).

Формат выходных данных

Для каждого из T тестовых примеров выведите в отдельной строке два числа: каким может оказаться минимальное и максимальное число решивших задачу.

Примеры

Входные данные

```
3
3 2 1
5 5 0
4 3 3
```

Выходные данные

```
1 2
5 5
0 1
```

"B" Сто

Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Все люди любят круглые числа. У этого явления достаточно много причин, главной из которых является то, что число, которое делится, например, на 100, существенно проще запомнить. К сожалению, в нашей жизни круглых чисел существенно меньше, чем всех остальных.

Первоклассник Петя очень переживает по этому поводу. Поэтому он придумал свое определение круглого числа.

По версии Пети, число является круглым только в том случае, если из него можно удалить ровно k цифр так, что после этого число не будет содержать ведущих нулей, но будет делиться на 100. Например, при k равным 2, число 10304 является круглым, а число 1000 — нет.

Теперь Петя хочет узнать, какие числа при каких k являются круглыми, а какие — нет. Помогите ему.

Формат входных данных Первая строка содержит целое число n — количество пар чисел k и x , про которые Петя хочет что-то узнать. Следующие n строк содержат по два целых положительных числа k и x каждая. Гарантируется, что любое число k строго меньше количества цифр в соответствующем числе x . Гарантируется, что все числа x не содержат ведущих нулей. Гарантируется, что сумма количеств цифр во всех числах x не превышает 10^5 .

Формат выходных данных Для каждой пары чисел k и x из входного файла выведите -1 в случае, если число x не является круглым при данном k . В случае же, если число x является круглым при данном k , выведите число, из которого вычеркнули ровно k цифр, после чего оно стало делиться на 100. Выведенное вами число не должно содержать ведущих нулей, но может быть равно нулю.

Russian Code Cup 2014
Квалификация 4

Примеры

Входные данные

3
2 10304
2 1000
3 1023

Выходные данные

100
-1
0

Russian Code Cup 2014
Квалификация 4

"С" Поход в гости

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

Члены жюри Russian Code Cup очень любят ходить друг к другу в гости. А как известно, никто не любит, когда к нему в гости приходят с пустыми руками. Поэтому, чтобы не обидеть хозяев, каждый порядочный человек, который идет в гости, берет с собой подарок.

Но никто не любит тратить деньги на покупку подарков. Поэтому, когда человек собирается в гости, он пойдет покупать новый подарок только в том случае, если у него не окажется подарков, которые дарили ему. Если же у него есть хотя бы один подарок, то он выберет из них тот, который ему подарили раньше всего, и пойдет в гости с ним.

К сожалению, у такой схемы есть свои недостатки. Когда человеку дарят подарок, который покупал он сам, он расстраивается, но все равно принимает этот подарок и угощает гостя вкусным чаем.

В жюри n человек, и они m раз ходили друг к другу в гости и дарили подарки. Про каждый такой поход вам необходимо сказать, расстроился ли хозяин дома из-за подарка или нет.

Формат входных данных

Первая строка содержит целое положительное число T — число тестовых примеров во входных данных.

Описание каждого тестового примера начинается с двух целых положительных чисел n и m , обозначающих количество членов жюри и количество походов в гости. В следующих m строках описан очередной поход в гости двумя числами a и b ($1 \leq a, b \leq n$). Они обозначают, что человек с номером a пришел в гости к человеку с номером b . Гарантируется, что никто не ходит в гости сам к себе. Гарантируется, что суммарное количество походов в гости во всех тестовых примерах не превышает 10^5 . Аналогично, общее количество людей не превышает 10^5 .

Russian Code Cup 2014
Квалификация 4

Формат выходных данных

Для каждого тестового набора выведите m слов. Про каждый поход в гости выведите YES, если хозяин расстроился и NO в противном случае.

Примеры

Входные данные

```
1
3 7
1 2
1 3
3 2
3 1
2 1
2 1
2 1
```

Выходные данные

```
NO
NO
NO
NO
YES
YES
NO
```

"D" СММ

Ограничение по времени

2 секунды

Ограничение по памяти

256 мегабайт

Как Вы уже могли заметить, в жюри Russian Code Cup любят обратные задачи. Вот еще одна!

Система непересекающихся множеств — это структура данных, которая позволяет хранить набор элементов, представленных в виде набора непересекающихся множеств. Она позволяет выполнять две операции:

- $\text{union}(a, b)$ — объединяет два множества, содержащие элементы a и b .
- $\text{find}(a)$ — возвращает некоторого представителя множества, в котором содержится a . $\text{find}(a) = \text{find}(b)$ тогда и только тогда, когда a и b содержатся в одном множестве.

Популярным подходом к реализации данной структуры данных является реализация в виде леса корневых деревьев. Каждое множество представляется в виде корневого дерева, представителем которого является корень дерева. Родителя вершины i обозначим $\text{parent}[i]$ (для корня дерева положим $\text{parent}[i] = i$). Изначально, каждый элемент находится в своем множестве и $\text{parent}[i] = i$ для всех i . Операция union в таком представлении реализуется, как подвешивание одного из корней дерева к другому. Операция find реализуется, как простой проход по ссылкам к родителю, пока не достигнут корень дерева. Для простоты будем считать, что аргументы процедуры union — корни различных деревьев.

Также в этой задаче используется ранговая эвристика: введем дополнительную величину $\text{rank}[i]$, которая изначально равна нулю для всех i . Операция $\text{union}(a, b)$ работает следующим образом: если $\text{rank}[a] = \text{rank}[b]$, то $\text{rank}[a]$ увеличивается на единицу. После этого вершина с меньшим rank подвешивается к вершине с большим rank .

Приведем псевдокод процедур union и find .

```
union(a, b)
    if rank[a] == rank[b] then
        rank[a] = rank[a] + 1
    if rank[a] > rank[b] then
        parent[b] = a
    else
        parent[a] = b
```

Russian Code Cup 2014
Квалификация 4

```
find(a)
    while parent[a] != a do
        a = parent[a]
    return a
```

Вам дан массив $parent[i]$. Существует ли такая последовательность операций *union*, что после ее выполнения массив $parent$ станет равен заданному?

Формат входных данных Первая строка содержит целое число n ($1 \leq n \leq 10^4$). Вторая строка содержит n целых чисел $parent[1], \dots, parent[n]$ ($1 \leq parent[i] \leq n$ для всех i от 1 до n).

Формат выходных данных Если требуемой последовательности не существует, выведите -1. Иначе в первой строке выведите целое число $k \geq 0$ — количество операций. Затем выведите k пар чисел $a_i b_i$ ($1 \leq a_i, b_i \leq n$) — аргументы i -й операции *union*. К моменту выполнения i -й операции a_i и b_i должны быть корнями различных деревьев.

Примеры

Входные данные

```
4
1 1 1 4
```

Выходные данные

```
2
1 2
3 1
```

Входные данные

```
3
2 3 3
```

Выходные данные

```
-1
```

"Е" Нанороботы

Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Глубоко в недрах одной секретной лаборатории ведутся разработки прототипа жидкого робота. Фактически, этот робот состоит из огромного количества управляемых независимо нанороботов, которые, координируя свои действия, образуют требуемую макроструктуру. Исследования пока далеки от окончания. Количество используемых вместе нанороботов невелико, да и алгоритмы перемещения оставляют желать лучшего.

Все роботы получают команды с сервера. К сожалению, серверный софт пока тоже не доработан. В частности, в каждый момент времени пользоваться сервером может только какой-то один робот. Второму подключившемуся придётся дождаться выполнения команды первого, чтобы выполнить свою. Таким образом общее время выполнения роботами некоторых действий равно суммарному числу команд, полученных ими от сервера.

Робот, состоящий из w нанороботов, изначально установлен в левом верхнем углу испытательного полигона размером $n \times m$ клеток. Его цель — добраться всеми своими нанороботами до правого нижнего угла испытательного полигона. Каждая клетка полигона имеет ограничение, какое максимальное число нанороботов может на ней одновременно находиться. Робот может послать команду серверу и получить команду движения на одну клетку в каком-то из четырёх направлений. Также, если робот в настоящий момент состоит из x нанороботов, он может получить команду разделиться на двух роботов, состоящих из u и v нанороботов, соответственно, где $u + v = x$. После этого эти два робота действуют независимо. Объединиться обратно роботы не могут.

Помогите проверить оптимальность используемого алгоритма перемещения — посчитайте, какое наименьшее количество команд сервер должен послать, чтобы все нанороботы переместились в правый нижний угол полигона.

Russian Code Cup 2014
Квалификация 4

Формат входных данных

В первой строке заданы целые числа n , m и w ($1 \leq n, m \leq 10$, $nm \geq 2$, $1 \leq w \leq 500$). Далее, в n строках задано по m **положительных** целых чисел $a_{i,j}$ — максимальное число нанороботов, которые могут находиться в клетке i -й строки j -го столбца ($a_{i,j} \leq 500$). **Гарантируется, что в начальной и конечной клетках могут находиться хотя бы w роботов.**

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

Входные данные

```
1 3 10
10 10 10
```

Выходные данные

```
2
```

Входные данные

```
1 3 10
10 5 10
```

Выходные данные

```
5
```

Входные данные

```
2 2 10
10 5
5 10
```

Выходные данные

```
5
```