

Задача А. Рейтинг

Имя входного файла: a.in
 Имя выходного файла: a.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Новый учитель математики ввел в школе оригинальную систему оценки учеников – рейтинговую. На каждом уроке школьнику предлагалось выполнить задание, состоящее из нескольких задач. После этого учитель увеличивал рейтинг школьника на число, равное отношению количества решенных им сегодня задач к количеству задач, решенных им на прошлом занятии. Например, если сегодня ученик решил 5 задач, а вчера – две, то к его рейтингу сегодня прибавится $\frac{5}{2} = 2.5$. Изначально рейтинг равен нулю; он начинает увеличиваться со второго дня. Если в один из дней какой-то ученик не решал ни одной задачи, то учитель объявлял его полной бездарностью и переводил в другой класс (с облегченным изучением математики), поэтому каждый день все ученики старались решить хотя бы одну задачу.

Школьники быстро обнаружили, что для получения наибольшего рейтинга им далеко не всегда нужно решать все задачи. Им известно, сколько задач учитель будет давать в каждый из дней. Помогите им определить, в какой день сколько задач следует решить, чтобы в итоге получить наибольший рейтинг.

Формат входных данных

В первой строке входного файла записано одно натуральное число N ($1 \leq N \leq 1\,000$) – количество уроков, которые провел в школе учитель до того, как его уволили.

В следующей строке записаны числа $a_1, a_2, \dots, a_N$ – количество задач, которые учитель предложил школьникам на первом, втором, ..., N -м уроках соответственно ($1 \leq a_i \leq 100$).

Формат выходных данных

В первой строке выведите максимальный рейтинг, который может получить школьник за N дней, с точностью не менее 0.001.

Во второй строке выведите N чисел – количество задач, которые должен решить школьник в каждый из дней. Если вариантов несколько, выведите один любой из них.

Пример

a.in	a.out
3 1 2 3	4 1 1 3
3 1 10 8	10.800000 1 10 8

Задача В. Олимпиада-2010

Имя входного файла: `b.in`
Имя выходного файла: `b.out`
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Команда Москвы на Всероссийской олимпиаде по информатике 2010 года оказалась настолько велика, что помещается только в 2 вагона. Некоторые школьники испытывают друг к другу такую личную неприязнь, что не могут кушать, находясь в одном вагоне. Поскольку не кушать в поезде нельзя (это противоречит СанПиНам), то участников олимпиады надо рассадить так, чтобы школьники, испытывающие взаимную неприязнь, ехали в разных вагонах. Если это невозможно, никто никуда не поедет.

Формат входных данных

Первая строка входного файла содержит количество школьников N ($1 \leq N \leq 10000$) и количество взаимных неприязней M ($1 \leq M \leq 100000$). Следующие M строк содержат пары чисел, задающих номера участников, испытывающих взаимную неприязнь. Участники нумеруются с единицы.

Формат выходных данных

В случае, если рассадить участников невозможно — выведите единственное число 0. Если же рассадка возможна, выведите в первой строке номера школьников, едущих в первом вагоне, во второй — едущих во втором вагоне. Первый школьник всегда должен ехать в первом вагоне. Школьник с меньшим номером должен находиться в первом вагоне, если это возможно. Номера школьников в каждом из вагонов должны быть упорядочены по возрастанию.

Пример

<code>b.in</code>	<code>b.out</code>
3 3 1 2 3 2 1 3	0
4 3 1 2 2 3 2 4	1 3 4 2

Задача С. Автобусные маршруты

Имя входного файла: c.in
 Имя выходного файла: c.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Для решения транспортной проблемы в некотором городе до недавнего времени использовались N ($N \leq 100000$) автобусных маршрутов. Каждый маршрут начинался на одной из M ($M \leq 100000$) площадей и там же заканчивался. В процессе проезда по маршруту автобус мог несколько раз проезжать одну и ту же площадь, и даже мог проезжать более одного раза по одной и той же улице в одном и том же направлении.

В определенный момент местные власти решили сократить количество автобусных маршрутов в городе до одного. По их мнению, должен был остаться лишь один маршрут, который проходил бы по всем улицам, по которым раньше проходили автобусные маршруты, причем в том же направлении (но не обязательно в том же порядке). Если по каким-либо улицам автобусы ездили в обоих направлениях, то и новый маршрут должен проходить по этим улицам в обоих направлениях. По тем улицам и в тех направлениях, по которым раньше автобусы не ездили, новый маршрут проходить не должен. Однако так как контролеров увольнять нельзя, власти решили, что по каждой улице в каждом направлении новый маршрут должен проходить столько раз, сколько по ней проходили все старые маршруты, вместе взятые.

Требуется написать программу, которая для заданных исходных данных определяет требуемый местным властям автобусный маршрут.

Формат входных данных

Входной файл состоит из следующей последовательности строк. Первая строка содержит число N – количество автобусных маршрутов, M – количество площадей. Каждая из последующих N строк служит для описания соответствующего автобусного маршрута и содержит сначала число k ($k \leq 100000$), определяющее количество элементов маршрута, а затем $k+1$ чисел, задающих номера площадей, которые последовательно проезжает автобус на этом маршруте. Общая длина маршрутов не более 100000 улиц. При описании маршрута всегда задаются номера первой и последней площади маршрута, причем они всегда совпадают.

Формат выходных данных

Выходной файл должен содержать либо описание нового маршрута в том же формате, что используется во входном файле, либо одно число 0, если организовать требуемый маршрут не удастся.

Пример

c.in	c.out
2 5	9 1 2 3 4 5 4 3 2 1
6 1 2 3 4 3 2 1	
2 4 5 4	