

Задача А. Гриб

Имя входного файла:	mushrooms.in
Имя выходного файла:	mushrooms.out
Ограничение по времени:	1.5 секунды
Ограничение по памяти:	256 мегабайт

В очень большом лесу раскинулась огромная поляна. После проливного дождя на поляне последовательно вырастают грибы в течении n минут. На первой минуте в центре поляны вырастает гриб высотой h_1 сантиметров. На второй минуте на расстоянии 1 метр вокруг первого гриба вырастают грибы высотой h_2 . На n -ой минуте на расстоянии 1 метр вокруг уже существующих грибов вырастают грибы высотой h_n . грибы растут кольцами.

Разумеется, что такой урожай не мог остаться незамеченным. Поэтому Филипп занялся исследованием высоты грибов, выросших после дождя. А именно, его интересует, какой самый высокий гриб находится в области между окружностями радиусов r и R , где r — радиус меньшей окружности, а R — радиус большей окружности, центры окружностей находятся в точке, где вырос первый гриб.

Так как поляна может быть достаточно большой, то Филипп должен проверить несколько колец. Причем, если на $i - 1$ -й проверке было выявлено, что самый высокий гриб имеет высоту H (в кольце от r до R), то на i -ой проверке необходимо выявить гриб в кольце от $\min(R, (r \cdot H + H^2) \bmod n)$ до $\max(R, (r \cdot H + H^2) \bmod n)$. Филипп хочет узнать только сумму высот грибов полученных, в результате всех проверок.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество минут, в течении которых вырастают грибы. Во второй строке задано n целых чисел h_i — высота грибов, выросших на i минуте ($1 \leq h_i \leq 10^9$). В третьей строке задано число m ($1 \leq m \leq 5 \cdot 10^6$) — количество проверок. Последняя строка содержит два целых числа r и R ($0 \leq r \leq R < n$) — описание первой проверки.

Формат выходных данных

Выведите ответ на задачу — сумму высот грибов, полученных в результате проверок.

Примеры

mushrooms.in	mushrooms.out
5 1 2 3 4 5 3 1 4	15
7 1 1 4 2 3 5 1 1 0 6	5

Задача В. Соединение точек

Имя входного файла: `connect.in`
Имя выходного файла: `connect.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны n точек на плоскости. Требуется провести отрезки между некоторыми парами точек таким образом, чтобы, во-первых, из любой данной точки в любую можно было пройти по этим отрезкам, а во-вторых, суммарная длина проведённых отрезков была минимальна.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 200$) — количество точек. Следующие n строк содержат по два целых числа x_i, y_i ($-1000 \leq x_i, y_i \leq 1000$).

Гарантируется, что никакие две точки не совпадают, никакие три не лежат на одной прямой.

Формат выходных данных

В первой строке выходного файла выведите l — суммарную длину проведённых отрезков с точностью не менее шести десятичных знаков после запятой. Во второй строке выведите k — их количество.

В следующих k строках выведите по два числа a_j, b_j — номера точек, соединённых j -м отрезком ($1 \leq a_j, b_j \leq n, a_j \neq b_j$). Точки нумеруются с единицы в том порядке, в котором они даны во входном файле. Если ответов с минимальным l несколько, разрешается выводить любой из них.

Примеры

<code>connect.in</code>	<code>connect.out</code>
4 0 0 0 1 1 0 1 1	3 3 1 2 2 4 4 3
5 0 0 0 2 1 1 3 0 3 2	7.064495 4 3 1 3 2 3 4 4 5

Задача С. Расстояние между вершинами

Имя входного файла: `distance.in`
Имя выходного файла: `distance.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит натуральные числа n , m , s и f ($n \leq 70000$, $m \leq 2 \cdot 10^5$, $1 \leq s, f \leq n$, $s \neq f$) — количество вершин и рёбер графа, а также номера вершин, длину пути между которыми требуется найти.

Следующие m строк содержат по три натуральных числа b_i , e_i и w_i ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 10^5$) — номера концов i -го ребра и его вес, соответственно.

Формат выходных данных

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами s и f . Во второй строке через пробел выведите вершины на кратчайшем пути из s в f в порядке обхода.

Если пути из s в f не существует, выведите -1 .

Примеры

<code>distance.in</code>	<code>distance.out</code>
4 4 1 3	3
1 2 1	1 2 3
3 4 5	
3 2 2	
4 1 4	

Задача D. Шоколадка

Имя входного файла: `choco.in`
Имя выходного файла: `choco.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Двое играют в такую игру: перед ними лежит шоколадка размера $n \times m$. Игроки ходят по очереди. За ход разломить любой имеющийся кусок шоколадки на 2 непустых куска, при этом запрещено ломать куски размером не больше, чем $1 \times s$ (т.е. нельзя ломать куски, у которых один размер равен 1, а другой не превосходит s), куски можно поворачивать. Ломать, конечно, можно только вдоль линий, нанесенных на шоколадке, т.е. после разлома должны получаться два прямоугольника с целочисленными ненулевыми сторонами. Проигрывает тот, кто не может сделать ход.

Формат входных данных

Во входном файле находятся три целых числа n, m, s ($1 \leq n, m, s \leq 100$).

Формат выходных данных

В выходной файл выведите 1 или 2 — номер игрока, который выигрывает при правильной игре.

Примеры

<code>choco.in</code>	<code>choco.out</code>
2 2 1	1

Задача E. Почтальон

Имя входного файла: `post.in`
Имя выходного файла: `post.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В городе есть n площадей, соединенных улицами. При этом количество улиц не превышает ста тысяч и существует не более трех площадей, на которые выходит нечетное число улиц. Для каждой улицы известна ее длина. По улицам разрешено движение в обе стороны. В городе есть хотя бы одна улица. От любой площади до любой можно прийти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице так, что бы длина его пути была наименьшей. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество площадей в городе ($1 \leq n \leq 1000$). Далее следуют n строк, задающих улицы. В i -ой из этих строк находится число m_i — количество улиц, выходящих из площади i . Далее следуют m_i пар положительных чисел. В j -ой паре первое число — номер площади, в которую идет j -ая улица с i -ой площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улиц с площади на нее саму.

Все числа во входном файле не превосходят 10^5 .

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте (считая первую и последнюю), а во вторую — номера площадей в порядке их посещения.

Если решений нет, выведите в выходной файл одно число -1 .

Если решений несколько, выведите любое.

Примеры

<code>post.in</code>	<code>post.out</code>
4	5
2 2 1 2 2	1 2 3 4 2 1
4 1 2 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	

Задача F. Расстояние между отрезками

Имя входного файла: distance5.in
Имя выходного файла: distance5.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Найдите расстояние между двумя отрезками.

Формат входных данных

Входной файл содержит восемь целых чисел — координаты четырёх концов двух отрезков.

Формат выходных данных

Выведите одно число — расстояние между отрезками с точностью не менее 10^{-6} .

Примеры

distance5.in	distance5.out
1 1 2 2	0.7071067812
2 1 3 0	

Задача G. Задача для Пискалова

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

По заданным a, b найдите такие целые x, y, d , что $ax + by = d$, где d — наибольший общий делитель a и b .

Формат входных данных

Входные данные состоят из нескольких тестов. Каждая строка содержит два натуральных числа a, b , разделенных пробелом ($a, b \leq 10^9$).

Формат выходных данных

Для каждой пары a и b в отдельной строке вывести три целых числа x, y и d , разделенных пробелом. Если искомым значений x и y несколько, то следует выводить такую пару, для которой $|x| + |y|$ минимально. Если и таких пар несколько, то вывести ту пару, в которой x минимально.

Примеры

stdin	stdout
4 6	-1 1 2
17 17	0 1 17

Задача N. Палиндромы

Имя входного файла: `palindrome.in`
Имя выходного файла: `palindrome.out`
Ограничение по времени: 2 секунда
Ограничение по памяти: 256 мегабайт

Формат входных данных

Входной файл содержит одну строку s длины $|s|$ ($1 \leq |s| \leq 10^5$).

Формат выходных данных

Выведите количество пар $1 \leq i < j \leq |s|$ таких, что подстрока $s_{i...j}$ является палиндромом.

Примеры

<code>palindrome.in</code>	<code>palindrome.out</code>
aaa	3
abba	2
омах	0

Задача I. Задача Иосифа

Имя входного файла: `joseph.in`
Имя выходного файла: `joseph.out`
Ограничение по времени: 2 секунда
Ограничение по памяти: 64 мегабайта

N людей стоят по кругу. Они начинают считать себя по часовой стрелке, счет ведется с единицы. Как только количество посчитанных достигает p , последний посчитанный (p -й) человек покидает круг, а процесс счета начинается со следующего за ним человека и вновь ведется с единицы.

Последний оставшийся в кругу выигрывает.

Можете ли вы посчитать, номер выигравшего человека в исходном кругу? Люди нумеруются числами от 1 до N по часовой стрелке, начиная с того самого человека, с которого начинался счет.

Формат входных данных

Входной файл содержит два целых числа N и p ($1 \leq n, p \leq 10^6$).

Формат выходных данных

Выведите номер выигравшего человека.

Примеры

<code>joseph.in</code>	<code>joseph.out</code>
3 4	2

Задача J. Декартово Дерево

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 2 секунда
Ограничение по памяти: 256 мегабайт

Вам даны пары чисел (a_i, b_i) . Вам необходимо построить декартово дерево, такое что i -ая вершина имеет ключи (a_i, b_i) , вершины с ключом a_i образуют бинарное дерево поиска, а вершины с ключом b_i образуют кучу.

Формат входных данных

В первой строк записано число n — количество пар. Далее следуют n ($1 \leq n \leq 50000$) пар (a_i, b_i) . Для всех пар $|a_i|, |b_i| \leq 30000$. $a_i \neq a_j$ и $b_i \neq b_j$ для всех $i \neq j$.

Формат выходных данных

Если декартово дерево с таким набором ключей построить возможно, выведите в первой строке YES, в противном случае выведите NO. В случае ответа YES, выведите n строк, каждая из которых должна описывать вершину. Описание вершины состоит из трёх чисел: номер предка, номер левого сына и номер правого сына. Если у вершины отсутствует предок или какой-либо из сыновей, то выводите на его месте число 0.

Если подходящих деревьев несколько, выведите любое.

Примеры

tree.in	tree.out
7	YES
5 4	2 3 6
2 2	0 5 1
3 9	1 0 7
0 5	5 0 0
1 3	2 4 0
6 6	1 0 0
4 11	3 0 0