

Задача А. Шашки

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

Петя любит играть с шашками на клеточной доске. Недавно он изобрёл новую игру: превратить одну красивую картинку, составленную из шашек, в другую красивую картинку. Петя допускает следующие ходы. Для одной из конфигураций:

	A		
	B	C	

	B		
	C		A

он может убрать шашку из клетки A и поставить шашки в клетки B и C или наоборот — убрать шашки из клеток B и C и поставить шашку в клетку A . (Эти конфигурации могут браться в любом месте доски, но не могут быть повернуты или отражены.) Запрещается делать ход, если придётся убрать шашку из пустой клетки или поставить в занятую. У Пети настолько большая доска, что вы можете считать её бесконечной во всех направлениях. Также можно полагать, что у Пети бесконечно много шашек.

Петя решил поделиться головоломкой со своим другом Васей. Он попросил преобразовать одну картинку в другую. Вася пробовал разные комбинации ходов на протяжении долгого времени. Несколько раз он даже получал что-то похожее на свою цель, но не ту же самую картинку. Тогда Вася решил, что разгадать головоломку невозможно. Но Петя сказал, что у него есть список ходов.

Вася просит вас написать программу, которая помогла бы ему решить Петину головоломку.

Формат входных данных

Первая строка содержит два разделённых пробелом целых числа n и m ($1 \leq n, m \leq 10$) — размеры первой картинки. Следующие n строк описывают картинку. Каждая строка содержит по m символов «.» или «#», где «.» означает, что клетка свободна, а «#» — что клетка занята шашкой. Далее следует описание второй картинки в таком же формате.

Гарантируется, что вторую картинку можно получить из первой, используя описанные ходы.

Формат выходных данных

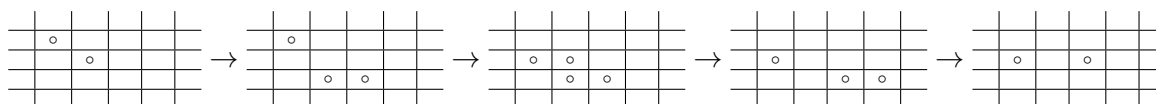
Выведите последовательность ходов, которая преобразует первую картинку во вторую. Каждая строка должна содержать 3 целых числа — координаты клетки A и номер конфигурации (1 или 2) следующего хода. Левая нижняя клетка картинки имеет координаты $(0, 0)$, а правая верхняя — $(m - 1, n - 1)$.

Примеры

стандартный ввод	стандартный вывод
2 2 #. .. 1 2 ##	0 1 1
2 2 #. .# 1 3 #.#	1 0 1 0 1 1 3 -1 2 2 0 1
2 1 # # 1 3 ..#	2 0 2

Замечание

Ниже проиллюстрирована последовательность ходов из второго примера.




Можно доказать, что для любого корректного набора входных данных существует последовательность не более чем из 70 000 ходов, которая позволяет решить головоломку.

Задача В. Ферзь атакует

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

Ферзь — самая сильная шахматная фигура. Ферзь может перемещаться на любое число свободных полей в любом направлении по прямой (по горизонтали, по вертикали и по диагонали), совмещая в себе возможности ладьи и слона.

	a	b	c	d	e	f	g	h	
8				×				×	8
7	×			×			×		7
6		×		×		×			6
5			×	×	×				5
4	×	×	×		×	×	×	×	4
3			×	×	×				3
2		×		×		×			2
1	×			×			×		1
	a	b	c	d	e	f	g	h	

Найдите на шахматной доске позицию, из которой ферзь атакует ровно k полей.

Формат входных данных

В единственной строке записано одно целое число k ($1 \leq k \leq 63$).

Формат выходных данных

Если подходящей позиции на шахматной доске нет, то выведите **Epic fail**. Иначе выведите позицию ферзя, находясь в которой, он атакует k **других** позиций шахматной доски. Сначала выведите обозначение вертикали — символ от **a** до **h**, затем номер горизонтали — число от 1 до 8. Если решений несколько, выведите любое.

Пример

стандартный ввод	стандартный вывод
27	d4

Задача С. Две окружности в трёхмерном пространстве

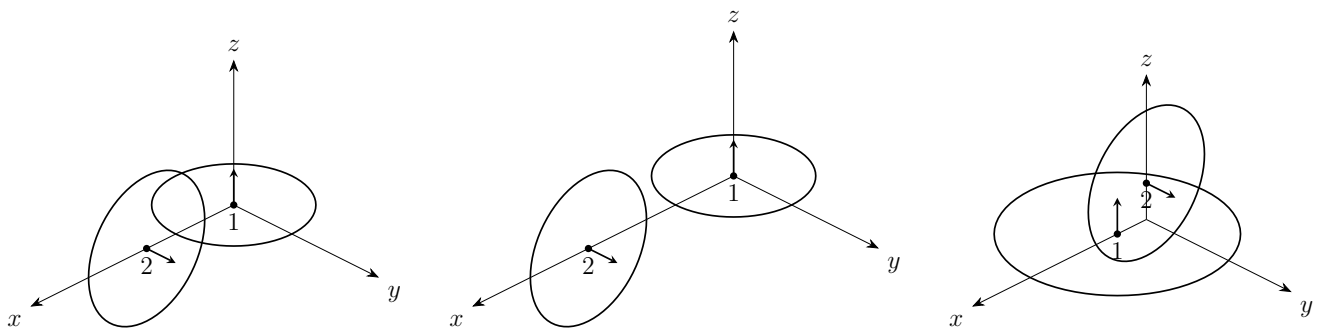
Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Вам даны две окружности в трёхмерном пространстве.

Для каждой окружности известны:

- координаты центра окружности;
- радиус окружности;
- вектор нормали к плоскости, в которой лежит окружность.

Определите, являются ли эти окружности зацепившимися, т. е. верно ли, что каждая из двух окружностей пересекает круг, ограниченный второй окружностью.



Формат входных данных

В первой строке записано целое число T ($1 \leq T \leq 100\,000$) — количество тестовых примеров.

Каждый тестовый пример описывается двумя строками:

- В первой строке через пробел записаны семь целых чисел $x_c^{(1)} y_c^{(1)} z_c^{(1)} r^{(1)} x_n^{(1)} y_n^{(1)} z_n^{(1)}$ — координаты центра, радиус и координаты вектора нормали первой окружности.
- Во второй строке через пробел записаны семь целых чисел $x_c^{(2)} y_c^{(2)} z_c^{(2)} r^{(2)} x_n^{(2)} y_n^{(2)} z_n^{(2)}$ — координаты центра, радиус и координаты вектора нормали второй окружности.

Все параметры окружностей в тестовом наборе не превосходят 10^4 по абсолютной величине. Радиусы окружностей положительны. У каждого вектора нормали хотя бы одна координата является ненулевой.

Гарантируется, что при изменении радиусов окружностей не более чем на 0,1 ответ на задачу не меняется.

Формат выходных данных

Выведите T строк. Для каждого тестового примера в порядке их следования выведите единственное слово в строке: YES, если окружности являются зацепившимися, или NO в противном случае.

Пример

стандартный ввод	стандартный вывод
3	YES
0 0 0 2 0 0 1	NO
3 0 0 2 0 1 0	NO
0 0 0 2 0 0 1	
5 0 0 2 0 1 0	
1 0 0 3 0 0 1	
0 0 1 2 0 1 0	

Задача D. Развязывающее множество

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

Ориентированный граф $G = (V, E)$ называется *сильно связным*, если из любой его вершины $a \in V$ можно попасть в любую вершину $b \in V$, следуя вдоль рёбер графа из множества E .

Будем называть множество E' *развязывающим множеством* графа G , если $E' \subseteq E$ и граф $G' = (V, E \setminus E')$ не является сильно связным. Иными словами, если после удаления всех рёбер из множества E' граф G больше не сильно связан, то множество E' является для него развязывающим.

По заданному графу G найдите его развязывающее множество **наименьшей мощности** либо определите, что такого множества нет.

Формат входных данных

В первой строке записано целое число T ($1 \leq T \leq 1000$) — количество тестовых примеров.

Каждый тестовый пример описывается $m + 1$ строкой. В первой строке через пробел записаны два целых числа n и m ($1 \leq n \leq 100$, $0 \leq m \leq n(n - 1)$) — количество вершин и количество рёбер графа соответственно. Далее в каждой из m строк находится по два целых числа a_i, b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — соответственно начало и конец i -го ориентированного ребра.

Рёбра нумеруются от 1 до m в том порядке, в котором они поступают на вход. Гарантируется, что в каждом тестовом примере никакая упорядоченная пара (a_i, b_i) не появляется дважды. Также гарантируется, что в каждом тесте находится не более чем 5 графов с $n > 50$, не более чем 20 графов с $n > 20$ и не более чем 100 графов с $n > 10$.

Формат выходных данных

Для каждого тестового примера выведите ответ на задачу.

Если развязывающего множества для графа не существует, выведите в строке число -1 .

В противном случае выведите в первой строке целое число k ($0 \leq k \leq m$) — наименьшая возможная мощность развязывающего множества. Если $k > 0$, то во второй строке выведите через пробел k номеров рёбер, входящих в искомое множество. Номера рёбер должны следовать по возрастанию.

Если есть несколько ответов, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
3	0
4 3	2
1 3	1 2
2 3	1
2 4	1
5 10	
1 2	
1 3	
2 3	
2 4	
3 4	
3 5	
4 1	
4 5	
5 1	
5 2	
2 2	
1 2	
2 1	

Задача Е. Приближение дроби

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

Египетские математики в 1800 году до н. э. записывали рациональные числа от 0 до 1 как сумму нескольких обратных различным целым. Например, число $\frac{2}{5}$ они могли записать как $\frac{1}{3} + \frac{1}{15}$ или $\frac{1}{4} + \frac{1}{10} + \frac{1}{20}$, но не как $\frac{1}{5} + \frac{1}{5}$.

Вам необходимо приблизить дробь $\frac{a}{b}$ суммой нескольких обратных различным целым. Ваше приближение $x = \sum \frac{1}{n_i}$ должно удовлетворять неравенству $\frac{a}{b} \leq x < \frac{a+1}{b}$.

В своём приближении вы можете использовать не более 1000 слагаемых, все используемые числа в разложении не должны превосходить 10^{18} . Гарантируется, что разложение, удовлетворяющее данным требованиям, всегда существует.

Формат входных данных

Единственная строка содержит два разделённых пробелом целых числа a и b ($1 \leq a < b \leq 10^{13}$). Гарантируется, что дробь $\frac{a}{b}$ является несократимой.

Формат выходных данных

В первой строке выведите целое число k ($1 \leq k \leq 1000$) — количество слагаемых.

Во второй строке выведите k попарно различных целых чисел от 1 до 10^{18} — знаменатели слагаемых, дающих в сумме приближение числа $\frac{a}{b}$.

Если число $\frac{a}{b}$ можно приблизить несколькими различными способами, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
2 5	2 3 15
1 10000000000000	1 10000000000000
3 7	3 5 6 7

Задача F. Уравнение

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

Дано n положительных целых чисел a_1, a_2, \dots, a_n .

Найдите число решений уравнения $a_1x_1 + a_2x_2 + \dots + a_nx_n = m$ в положительных целых числах x_1, x_2, \dots, x_n .

Так как искомое число решений может быть слишком большим, найдите его остаток от деления на 998 244 353.

Формат входных данных

В первой строке через пробел записаны два целых числа n и m ($1 \leq n \leq 100$, $0 \leq m \leq 2^{63} - 1$).

Во второй строке записаны n целых положительных чисел a_1, a_2, \dots, a_n ($\sum_{i=1}^n a_i \leq 100$).

Формат выходных данных

Выведите остаток от деления искомого числа решений уравнения на 998 244 353.

Примеры

стандартный ввод	стандартный вывод
4 21 1 2 3 4	27
4 2019 21 9 20 19	18142
5 6 1 1 1 1 1	5

Задача G. Скользящая медиана

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

Медиана в математической статистике — число, характеризующее выборку (например, набор чисел). Медиану можно найти, упорядочив элементы выборки по неубыванию и взяв средний элемент. Например, выборка (11, 9, 3, 5, 5) после упорядочивания превращается в (3, 5, 5, 9, 11), и её медианой является число 5. Если в выборке чётное число элементов, в качестве медианы будем использовать полусумму двух соседних значений (то есть медиану набора (1, 3, 5, 7) считаем равной 4).

На вход поступают запросы двух видов: на добавление одного числа в выборку и на вычисление медианы. Нужно научиться эффективно их обрабатывать.

Формат входных данных

Целые числа (от 0 до 10^{18}) записаны в одну строку. Если число положительное, то его нужно добавить в выборку. Если число равно нулю, то нужно вычислить медиану выборки, которая была накоплена к данному моменту. Общее количество чисел на входе не менее 1 и не превосходит 300 000. Гарантируется, что первое число не равно нулю.

Формат выходных данных

Для каждого числа 0 на входе выведите в отдельной строке полученное значение медианы с абсолютной погрешностью не более 0,1.

Примеры

стандартный ввод	стандартный вывод
1 0 2 0 3 0	1 1.5 2
11 9 3 5 5 0	5
1 3 5 7 0	4

Задача Н. Тридцатиугольник

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

Вершины правильного тридцатиугольника пронумерованы числами от 1 до 30 против часовой стрелки. Выберите k вершин таким образом, чтобы центр масс этих точек совпадал с центром тридцатиугольника.

Формат входных данных

В единственной строке входных данных записано одно целое число k ($1 \leq k \leq 30$).

Формат выходных данных

Если подходящего набора вершин не существует, выведите строку **Epic fail**.

Иначе выведите строку из 30 символов. i -й символ должен быть равен '1', если i -я вершина была выбрана, или '0' в противном случае.

Если подходящих наборов вершин несколько, выведите любой из них.

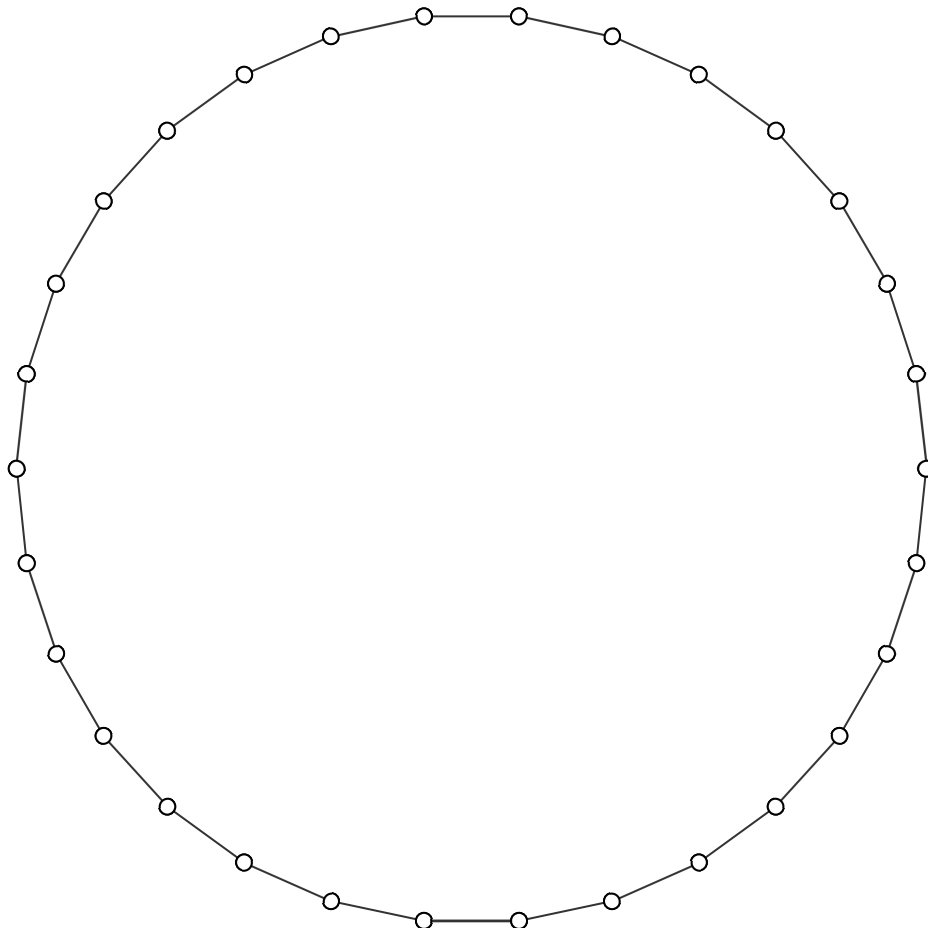
Пример

стандартный ввод	стандартный вывод
9	111000000011100000001110000000

Замечание

Центром масс системы точек $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ на плоскости является точка с координатами $(\frac{\sum x_i}{k}, \frac{\sum y_i}{k})$.

Вы можете использовать рисунок, чтобы лучше представить решение:



Задача I. Прогрессия

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

Задан массив A длины n . Требуется изменить его элементы так, чтобы они образовали арифметическую прогрессию. Каждый элемент массива можно изменить не более одного раза: или увеличить на единицу, или уменьшить на единицу. Какое наименьшее число элементов нужно модифицировать?

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 10^5$).

Во второй строке через пробел записано n целых чисел a_i ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — минимально возможное число операций. Если получить арифметическую прогрессию указанным способом невозможно, выведите **Epic fail**.

Примеры

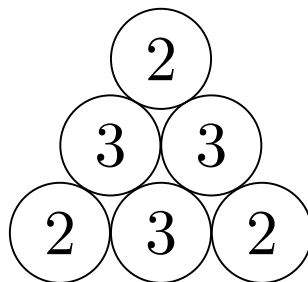
стандартный ввод	стандартный вывод
2 300 300	0
5 1 3 6 9 12	1
5 1 3 6 3 6	Epic fail

Задача J. Шарики с числами

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

Костя расположил в форме треугольника $3k - 3$ шарика, каждая сторона треугольника состоит ровно из k шариков. На одной стороне треугольника находятся шарики с номерами $1, 2, \dots, k$, на второй — шарики с номерами $k, k + 1, \dots, 2k - 1$, на третьей — шарики с номерами $2k - 1, 2k, \dots, 3k - 3, 1$.

Косте нужно написать на шариках по одному целому числу из диапазона от 1 до 10^9 так, чтобы сумма чисел на шариках на каждой стороне треугольника была равна n и максимальное использованное число было как можно меньше. При этом Костя может использовать одинаковые числа сколько угодно раз. Если нельзя написать числа на шариках таким образом, то подскажите ему об этом.



Формат входных данных

В единственной строке через пробел записаны два целых числа k и n ($2 \leq k \leq 100$, $1 \leq n \leq 10^9$).

Формат выходных данных

Если подходящего способа написать числа на шариках нет, выведите **Epic fail**. Иначе выведите $3k - 3$ целых числа, которые Костя может написать на шариках, в порядке их нумерации. Если наборов чисел, удовлетворяющих условию задачи, несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
3 7	2 3 2 3 2 3
4 1	Epic fail

Задача К. Дороги

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

В Бйитландии n городов. Некоторые пары городов соединены между собой двусторонними дорогами. Сеть дорог позволяет из любого города проехать в любой другой. К сожалению, из-за недостатка финансирования дороги год за годом приходят в упадок и выводятся из эксплуатации. Известен порядок, в котором будут закрываться все дороги. Определите, в какой момент при закрытии очередной дороги жители Бйитландии потеряют возможность проехать из какого-либо одного города в другой город.

Формат входных данных

В первой строке через пробел записаны числа n и m ($1 \leq n \leq 100\,000$, $1 \leq m \leq 300\,000$) — число городов и число дорог соответственно.

В последующих m строках заданы все дороги Бйитландии в хронологическом порядке их закрытия. Дорога описывается парой чисел u и v ($1 \leq u, v \leq n$, $u \neq v$). Между парой городов может быть построено несколько дорог.

Формат выходных данных

Выведите одно число — номер дороги, после закрытия которой дорожная сеть утратит связность. Дороги нумеруются начиная с единицы в соответствии с порядком следования на входе.

Примеры

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

Задача L. Pixels per Inch

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

Задано разрешение экрана в пикселях и физический размер его диагонали в дюймах. Требуется вычислить разрешающую способность матрицы в точках на дюйм (англ. *ppi* — *pixels per inch*). Можно считать, что пиксель имеет форму квадрата.

Формат входных данных

Разрешение записано как произведение $w \times h$ ($1 \leq w, h \leq 10\,000$, числа w и h целые, разделителем является строчная латинская буква *x*). Длина диагонали d записана в виде вещественного числа с не более чем одним знаком после точки ($0.1 \leq d \leq 100$), после числа идёт символ дюйма в виде двойной кавычки (*"*).

Формат выходных данных

Выведите ответ, округлённый до ближайшего целого (в случае неоднозначности можно округлять в любую сторону).

Примеры

стандартный ввод	стандартный вывод
1920x1080 42"	52
240x320 3.5"	114