

Задача А. Паттерны

Имя входного файла: `patterns.in`
Имя выходного файла: `patterns.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Паттерном будем называть последовательность символов алфавита, символов ‘*’ и ‘?’ . *Алфавитом* в данной задаче будет множество строчных латинских букв.

Будем говорить, что строка s *удовлетворяет* паттерну a , если ее можно получить, заменив в a символы ‘*’ на некоторые (возможно пустые) последовательности символов алфавита, а символы ‘?’ на какие-то одиночные символы (к примеру, строка `abacaba` удовлетворяет паттерну `a*b?ca?*`).

Маленький Петя собрал солидную коллекцию паттернов, некоторые из которых даже являются палиндромами. Коллекция Пети растет с каждым днем, и скоро ему уже будет негде ее хранить. Поэтому Петя задумался о том, как бы уменьшить размеры своей коллекции. Он заметил, что у него скопилось уж очень много похожих паттернов, поэтому решил объединить некоторые из них.

Объединение паттернов a и b Петя определил так: если строка s удовлетворяет a или b , то она должна удовлетворять и их объединению. При этом среди всех возможных вариантов объединения выбирается паттерн с наименьшим количеством ‘*’. Если таких вариантов несколько, то из всех них выбирается паттерн с наибольшей длиной, если же при этом все еще остаются равноправные варианты, то из них выбирается любой вариант с меньшим числом символов ‘?’ . Петя хочет, чтобы Вы написали программу, которая автоматизирует процесс нахождения объединения двух паттернов. Помогите ему!

Формат входного файла

В первой строке входного файла записана строка с паттерном a . Во второй строке входного файла записана строка с паттерном b . Длины строк a и b не превышают 1000 символов.

Формат выходного файла

В первой строке выходного файла выведите объединение паттернов a и b , удовлетворяющее условию задачи.

Примеры

patterns.in	patterns.out
borland berland	b?rland
warrior war	war*
acm* ah?	a??*

Задача В. Квадродерево

Имя входного файла: `quadtree.in`
Имя выходного файла: `quadtree.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Квадродерево — это структура данных, которая может описывать, например, раскраску некоторой квадратной области. Квадродерево определяется рекурсивно следующим образом:

- Квадродерево считается целиком описанным (оно должно соответствовать квадратной области, закрашенной в один цвет). Тогда описание квадродерева состоит из цвета этой области;
- В противном случае описание квадродерева представляется в виде объединения описаний четырех квадродеревьев, соответствующих четырем областям, получаемым в результате деления данной квадратной области на четыре равные квадратные под-области.

Будем представлять квадродерево строкой, построенной на основе следующей грамматики:

$Q \rightarrow C$
 $Q \rightarrow + Q_{\text{верхнее левое}} Q_{\text{верхнее правое}} Q_{\text{нижнее левое}} Q_{\text{нижнее правое}}$
 $C \rightarrow a$
 $C \rightarrow b$
 \dots
 $C \rightarrow z$

Здесь Q — нетерминальный символ, соответствующий квадродереву, C — нетерминальный символ, соответствующий цвету области.

Пусть задано описание некоторого квадродерева в виде строки s . Будем считать, что самый маленький квадрат в квадродереве имеет сторону, равную 1. Вам необходимо найти площадь наибольшей связной области одного цвета. Область считается связной, если для любых двух ее точек существует кривая, соединяющая эти две точки и лежащая строго внутри области.

Формат входного файла

В первой строке входного файла записана строка s с описанием квадродерева. Длина строки s не превышает 100 000 символов.

Формат выходного файла

В первой строке выходного файла выведите площадь наибольшей связной области одного цвета. Во второй строке выходного файла выведите символ, соответствующий цвету искомой области. Если таких областей несколько, выведите символ, меньший в лексикографическом порядке. Гарантируется, что искомая площадь не превысит $2^{31} - 1$.

Примеры

quadtree.in	quadtree.out
a	1 a
+abaa	3 a
+a+abcd+dcbaa	5 a

Задача С. Многоугольник на шахматной доске

Имя входного файла: `polygon.in`
Имя выходного файла: `polygon.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На бесконечную шахматную доску, состоящую из квадратов со стороной, равной a , бросают правильный $4n$ -угольник диаметра $2b$ (радиус описанной вокруг него окружности равен b). Считается, что после броска все координаты центра многоугольника равновероятны, также как и любой угол поворота многоугольника вокруг его центра.

Определите вероятность того, что многоугольник пересечет сторону какого-либо квадрата.

Формат входного файла

В первой строке входного файла содержатся три целых числа a , b , n ($1 \leq a, b \leq 100, 1 \leq n \leq 10$).

Формат выходного файла

В первой строке выходного файла выведите искомую вероятность с точностью 100 знаков после запятой. Ваш ответ не должен отличаться от правильного более чем на 10^{-100} .

Пример

polygon.in
4 1 1
polygon.out
0.695738844611584016707573093204834 016473443270849924011976037992020103 319019330774530258763709557980

Пояснение: выходной файл должен представлять собой одну строку. Для удобства восприятия в приведенном примере она была разделена на три части.

Задача D. Небесная точка равновесия

Имя входного файла: `equilibrium.in`
Имя выходного файла: `equilibrium.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Известная страна Флатландия решила запустить спутник. В силу мироустройства Флатландии все планеты местной “солнечной системы” являются окружностями. Было решено поставить спутник в такую точку плоскости, чтобы все планеты были видны со спутника под одинаковым углом. Естественно, спутник не должен лежать внутри или на границе планеты. Обзор спутником каждой из планет не должен быть закрыт частично или полностью другими планетами.

Формат входного файла

В первой строке входного файла содержится число n ($2 \leq n \leq 4$) — число планет в солнечной системе. Далее следуют n строк по три числа в каждой строке x, y, r . Все числа целые и не превышают 100 по абсолютному значению. Планеты не могут перекрываться и касаться.

Формат выходного файла

Если такая точка для спутника существует, то выведите YES в первой строке выходного файла. В таком случае вторая строка выходного файла должна содержать два вещественных числа — координаты точки спутника. Разница между максимальным углом обзора и минимальным не должна превышать 10^{-4} . Если таких точек нет, то выведите в первой строке выходного файла NO.

Примеры

equilibrium.in	equilibrium.out
2 2 2 1 5 2 1	YES 3.5 0.0
3 0 4 1 4 4 1 4 0 1	YES 2.0 2.0
3 0 4 1 4 4 2 4 0 1	NO

Задача Е. Большой Шахматный Ним

Имя входного файла: `bignim.in`
Имя выходного файла: `bignim.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В шахматной стране в последнее время стала очень популярна игра Ним. Правила игры просты. В начальный момент игры на стол выкладываются несколько кучек камней. Два игрока ходят по очереди и за каждый ход берут из одной любой кучки произвольное число камней. Игрок, который не может сделать ход (больше не осталось камней), проигрывает.

Черный и белый короли тоже решили сыграть в Ним, но игра оказалась слишком сложной, поэтому они решили немного изменить правила: камни теперь можно брать не из любой кучки, а только из такой, в которой содержится максимальное число камней.

Так как все время первым ходит белый король, то ему стало интересно, а можно ли по начальной позиции определить, кто выиграет при оптимальной стратегии обоих игроков.

Формат входного файла

В первой строке входного файла содержится количество кучек n ($1 \leq n \leq 100$). Во второй строке содержатся n целых чисел a_i — количество камней в i -ой кучке ($1 \leq a_i \leq 100$).

Формат выходного файла

В первую строку выходного файла выведите White wins, если выиграет белый король. В противном случае выведите Black wins.

Примеры

bignim.in	bignim.out
2 1 3	White wins
2 1 1	Black wins

Задача F. Четная страна

Имя входного файла: `evencountry.in`
Имя выходного файла: `evencountry.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В шахматной стране есть n городов, соединенных m дорогами, причем два города могут соединяться более, чем одной дорогой, дорога может начинаться и кончаться в одном и том же городе, и из каждого города можно добраться в каждый.

Белый шахматный король совсем одурел и хочет, чтобы отныне из каждого города выходило четное число дорог (возможно, ни одной), причем добавлять новые дороги нельзя. Также у себя во дворце над

троном он хочет повесить табличку с числом способов проведения такой реорганизации.

Изготовление такой таблички поручили именно Вам.

Формат входного файла

В первой строке входного файла содержатся целые числа n и m ($1 \leq n \leq 1000$, $0 \leq m \leq 10000$). В последующих m строках содержится описание дорог, причем одна строка соответствует одной дороге. Каждая дорога описывается парой целых чисел a_i и b_i , разделенных пробелом. Такая запись означает, что i -ая дорога соединяет города с номерами a_i и b_i ($1 \leq a_i, b_i \leq n$).

Формат выходного файла

В первой строке выходного файла выведите число способов реорганизации дорог указанным в задаче образом.

Примеры

evencountry.in	evencountry.out
4 4 1 2 2 3 3 4 4 2	2
2 3 1 1 1 2 1 2	4

Задача G. Объем

Имя входного файла: `volume.in`
Имя выходного файла: `volume.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам необходимо найти объем простого многогранника, который задан своими гранями. Каждая грань представляет собой невырожденный треугольник и описывается координатами трех его вершин. Любое ребро многогранника принадлежит в точности двум его граням. Также многогранник не содержит самопересечений, самокасаний и его поверхность односвязна.

Формат входного файла

В первой строке входного файла содержится число граней многогранника n ($4 \leq n \leq 10000$). В последующих n строках содержится описание граней. Описание i -ой грани состоит из девяти чисел $x_{A_i} y_{A_i} z_{A_i} x_{B_i} y_{B_i} z_{B_i} x_{C_i} y_{C_i} z_{C_i}$, разделенных пробелами и представляющих собой координаты трех вершин грани $A_i B_i C_i$. Все координаты целые и по абсолютному значению не превышают 1000.

Формат выходного файла

В первой строке выходного файла выведите объем заданного многогранника — вещественное число с точностью 8 знаков после запятой. Выведенный объем не должен отличаться от правильного более чем на 10^{-8} .

Пример

volume.in	volume.out
4 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1	0.16666667

Задача H. Степень

Имя входного файла: `power.in`
Имя выходного файла: `power.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Определим $f(x)$ как функцию, вычисляющую сумму цифр числа x . Например, $f(239) = 2 + 3 + 9$. Рассмотрим функцию $g(n) = \sum_{i=2}^{10^n} f(i)$. Вам необходимо по заданному d найти такие числа x и k , что d^x делит $g(k)$, $1 \leq k \leq n$ и x максимально.

Формат входного файла

В первой строке входного файла содержатся целые числа n и d ($1 \leq n \leq 10^9$, $2 \leq d \leq 1000$).

Формат выходного файла

В первой строке выходного файла выведите требуемые числа x и k , разделенные пробелом. Если удовлетворяющих условию чисел k может быть несколько, выведите любое из них.

Примеры

power.in	power.out
2 10	2 2
1 2	0 1

Задача I. Сеть

Имя входного файла: **network.in**
Имя выходного файла: **network.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Рассмотрим ориентированный граф с n вершинами и m ребрами с истоком в вершине с номером 1 и стоком в вершине с номером n . При этом, гарантируется, что для любого ребра графа существует маршрут из истока в сток, проходящий по данному ребру.

Маршрутом называется чередующаяся последовательность вершин и ребер

$$v_0, e_1, v_1, \dots, v_{k-1}, e_k, v_k$$

в которой ребро $e_i = v_{i-1}v_i$. Такой маршрут называют маршрутом из вершины v_0 в вершину v_k .

Определим *стоимость* маршрута как сумму исходящих степеней всех его вершин. *Покрытием* ориентированного графа назовем такое множество маршрутов из истока в сток, что любое ребро данного графа принадлежит хотя бы одному из них. Стоимостью покрытия будем называть сумму стоимостей всех входящих в него маршрутов.

В данной задаче Вам необходимо найти покрытие минимальной стоимости.

Формат входного файла

В первой строке входного файла содержатся два целых числа n и m ($2 \leq n \leq 200$, $1 \leq m \leq 1000$) — число вершин и ребер графа, соответственно. На последующих m строках содержатся описания ребер, причем одна строка соответствует одному ребру. Каждое ребро описывается парой целых чисел a_i и b_i , разделенных пробелом. Такая запись означает, что i -ое ребро исходит из вершины a_i и входит в вершину b_i .

Формат выходного файла

В выходной файл выведите искомое покрытие минимальной стоимости в следующем виде. В первой строке выходного файла выведите стоимость минимального покрытия. Во второй строке выведите l — количество маршрутов в требуемом покрытии. В следующих l строках выведите искомые маршруты по одному на строке в следующем формате: первое число h ($h > 1$) определяет количество вершин в маршруте, далее следуют h чисел — номера вершин маршрута в порядке их прохождения.

Примеры

network.in	network.out
4 4	7
1 2	1
2 3	6 1 2 3 2 3 4
3 2	
3 4	
4 7	17
1 2	2
1 4	7 1 2 3 2 3 3 4
2 3	2 1 4
2 3	
3 2	
3 3	
3 4	

Задача J. Полотно

Имя входного файла: **painting.in**
Имя выходного файла: **painting.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Известный всему миру художник Гоги Ван Малевадзе очень любит абстрактную живопись. Его любимый инструмент рисования — Coral

Drow Abstractionist's Edition. В этом современном графическом редакторе остались только жизненно важные для настоящего абстракциониста функции — нарисовать отрезок и закрасить указанную область, ограниченную отрезками, в черный цвет.

Последнее творение художника — это замкнутая ломаная, нарисованная на бесконечном холсте так, что не существует такой точки, чтобы она принадлежала более чем двум ее звеньям. Ломаная разбивает холст на области, ограниченные ее звеньями. Внешнее по отношению к ломаной пространство считается бесконечной областью. Гоги считает две области *смежными*, если существует отрезок ненулевой длины, который целиком принадлежит обеим их границам.

Теперь художник хочет завершить свою работу и закрасить некоторые области черной краской так, чтобы из любых двух смежных областей ровно одна оказалась покрашенной в черный цвет. Гоги интересуется вопрос — какой суммарной площади окажутся все покрашенные в черный цвет области. От ответа на этот вопрос зависит, сколько краски для принтера ему надо купить. Естественно, он хочет потратить как можно меньше денег, поэтому его интересует раскраска с минимально возможной суммарной площадью черного цвета. Помогите ему!

Формат входного файла

В первой строке входного файла записано число n — количество звеньев ломаной ($3 \leq n \leq 500$). В последующих n строках содержатся вершины ломаной, перечисленные в порядке некоторого обхода, по одной на строку. Вершина описывается парой координат $v_i = (x_i, y_i)$. Это означает, что для любого $i \in [2..n]$, отрезок $v_{i-1}v_i$ — звено ломаной. Также звеном ломаной является отрезок v_nv_1 . Все координаты во входном файле — целые числа, не превышающие по абсолютному значению 1000.

Формат выходного файла

В первую строку выходного файла выведите минимально возможную суммарную площадь холста, которую Гоги надо закрасить в черный цвет, чтобы достичь желаемой раскраски. Выведенное число не должно отличаться от правильного больше чем на 10^{-4} . В случае, если требуемой раскраски не существует, выведите в качестве площади -1.

Примеры

painting.in	painting.out
3	0.5000
0 0	
1 1	
1 0	
4	6.0000
0 0	
2 0	
2 6	
4 4	

Задача K. Ломаная

Имя входного файла: **polyline.in**
Имя выходного файла: **polyline.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Недавно в школе Пете задали следующую задачу. По заданному числу n построить ломаную, покрывающую все точки из множества $S_n = \{(x, y) \mid x, y - \text{целые}, 0 \leq x < n, 0 \leq y < n\}$.

Недолго думая, Петя построил требуемую ломаную следующим образом: вертикальные звенья Петинной ломаной образуют множество прямолинейных отрезков

$$\{(i, 0) \rightarrow (i, n-1), 0 \leq i < n\}$$

а горизонтальные звенья — множество отрезков

$$\{(2i, 0) \rightarrow (2i+1, 0), 2i+1 < n, i \geq 0\} \cup$$

$$\{(2j+1, n-1) \rightarrow (2j+2, n-1), 2j+2 < n, j \geq 0\}$$

Однако учитель решил усложнить задачу, потребовав, чтобы количество звеньев у ломаной было меньше, чем у ломаной, построенной Петинным способом. Помогите Пете!

Формат входного файла

В первой строке входного файла содержится одно целое число n ($3 \leq n \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите количество звеньев ломаной k . В следующих $k + 1$ строках выведите информацию об узлах ломаной. В $(i + 1)$ -ой строке выведите координаты i -ого узла ломаной — два целых числа (x_i, y_i) , не превышающие по абсолютному значению 1000.

Пример

polyline.in	polyline.out
5	8 3 1 3 4 -1 0 4 0 4 4 -2 4 2 0 2 3 0 3

Задача L. Граница

Имя входного файла: `frontier.in`
Имя выходного файла: `frontier.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Во Флатландии начались беспорядки. Пойман известный разбойник и террорист Бен Бладен, который снова совершил ужасное злодеяние. В частности, преступнику инкриминируется многократный несанкционированный переход границы и незаконное распитие пива на чужой территории. Но, несмотря на это, широкая общественность недовольна ходом расследования и желает, чтобы преступнику было предъявлено более серьезное обвинение. Ключевым фактором в этом процессе явилось бы доказательство того, что преступник проходил через одну и ту же точку как минимум дважды.

Следствию известны координаты всех переходов границы в том порядке, в котором преступник их осуществлял. Также известно, что траектория следования преступника непрерывна и из любой ее точки можно попасть в любую другую. Граница разделяет два государства и представляет собой ось абсцисс. Уголовный кодекс Флатландии под переходом границы понимает строгое пересечение траекторией преступника числовой оси, ее касание переходом через границу не является.

Помогите следствию определить, можно ли из известной информации сделать вывод о том, что преступник прошел через одну и ту же точку дважды.

Формат входного файла

В первой строке входного файла содержится число точек перехода границы n ($1 \leq n \leq 50000$). Во второй строке содержатся n различных чисел a_i , записанных через пробел — координаты переходов границы преступником в том порядке, в котором они были совершены. Все числа во входном файле целые и не превышают по модулю 10^9 .

Формат выходного файла

В первую строку выходного файла выведите YES, если можно утверждать, что преступник прошел через одну и ту же точку дважды. В противном случае выведите NO.

Примеры

frontier.in	frontier.out
4 1 3 2 4	YES
4 1 2 3 4	NO

Задача M. Настольный теннис

Имя входного файла: `pingpong.in`
Имя выходного файла: `pingpong.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Петя и Миша играли в настольный теннис. В некоторый момент счет в партии был $a_1:b_1$, однако через некоторое время он стал $a_2:b_2$. Известно, что Петя каждый розыгрыш выигрывает с вероятностью p ($0 \leq p \leq 1$). Петя хочет узнать, увеличил ли он свои шансы на победу.

Партия в настольный теннис состоит из розыгрышей, каждый из которых заканчивается победой одного из игроков и приносит победителю 1 очко. В общем случае, партия играется до 11 очков. В случае

ничьи 10:10, партия продлевается до 12 очков. В случае ничьи 11:11 - до 13 очков. Более формально, игра прекращается, когда один из игроков набирает не менее 11 очков, при этом разница в очках составляет не менее двух очков. Такой игрок объявляется победителем партии.

Формат входного файла

В первой строке входного файла записаны пять чисел, разделенных пробелом: a_1 , b_1 , a_2 , b_2 и p . Гарантируется, что $a_1:b_1$ и $a_2:b_2$ — корректный счет, причем $a_1 \leq a_2 \leq 11$, $b_1 \leq b_2 \leq 11$.

Формат выходного файла

В первой строке выходного файла выведите YES, если Петя увеличил свои шансы на победу, и NO в противном случае.

Примеры

pingpong.in	pingpong.out
2 0 3 0 0.54	YES
6 6 8 8 0.5	NO