Задача A. Красивые числа

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

Будем называть число *красивым*, если сумма его цифр в десятичной системе счисления делится на количество цифр в нем (в десятичной системе счисления).

Необходимо найти n-ое в порядке возрастания красивое число.

Формат входного файла

Входной файл содержит целое число n ($1 \le n \le 100000$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

beauty.in	beauty.out
1	1
15	20

Задача B. Числа Фибоначчи

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

Числа Фибоначчи — это последовательность целых чисел, заданная рекуррентным соотношением: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$, $n \geq 2$.

Ваша задача — найти наибольший общий делитель двух чисел Фибоначчи.

Формат входного файла

Во входном файле два числа i и j ($1 \leq i, j \leq 10^6$) — номера чисел Фибоначчи.

Формат выходного файла

В выходной файл выведите остаток от деления наибольшего общего делителя чисел F_i и F_j на 10^9.

Примеры

fib.in	fib.out
5 10	5
2 4	1

Задача C. Развлечения с измерителем — 2

Имя входного файла:	`fun2.in`
Имя выходного файла:	`fun2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дима обнаружил у папы на столе специальный чертежный прибор, похожий на циркуль — измеритель. Измеритель отличается от обычного циркуля тем, что в обеих его ножках находятся иголки (у обычного циркуля в одной ножке находится иголка, а в другой — грифель).

Кроме измерителя Дима нашел на столе клетчатый лист бумаги, в углах некоторых клеток которого были нарисованы точки. Так как измеритель служит для измерения расстояний, то Дима решил измерить все попарные расстояния между всеми точками на листе бумаги.

Ваша задача — написать программу, которая по координатам точек определит, сколько различных расстояний встречается среди расстояний, которые измерил Дима.

Формат входного файла

Первая строка входного файла содержит число n — количество точек ($2 \le n \le 50$). Следующие n строк содержат по два целых числа — координаты точек. Координаты не превышают 10^4 по абсолютной величине.

Формат выходного файла

На первой строке выходного файла выведите k — количество различных расстояний, которые измерил Дима. Следующие k строк должны содержать по одному вещественному числу — сами расстояния. Расстояния должны быть выведены в возрастающем порядке. Каждое число должно быть выведено с точностью не менее чем 10^{-9}.

Примеры

fun2.in	fun2.out
4 0 0 1 1 1 0 0 1	2 1.0 1.414213562373

Задача D. Генерация тестов

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

При подготовке задач для олимпиад по информатике и программированию часто возникает необходимости подготовки тестов. Поскольку зачастую размеров тестов достаточно велик, то генерацию тестов разумно автоматизировать.

В геометрических задачах часто требуется сгенерировать n точек на плоскости так, чтобы никакие три из них не лежали на одной прямой. В этом и состоит ваша задача. Напишите программу, которая по числу n построит множество из n точек, обладающее указанным свойством.

Формат входного файла

Входной файл содержит целое число n ($1 \le n \le 300$).

Формат выходного файла

Если искомое множество точек можно построить, то выведите в первой строке выходного файла слово `YES`, а далее — n строк, каждая из которых должна содержать два числа — координаты соответствующей точки. Среди точек не должно быть совпадающих. Все координаты должны быть целыми числами, не превосходящими 10000 по абсолютному значению. Если искомое множество точек нельзя построить, выведите в выходной файл строку `NO`.

Примеры

gentest.in	gentest.out
1	YES 0 0
4	YES 0 0 0 1 1 0 1 1

Задача E. Левая рекурсия

Имя входного файла:	leftrec.in
Имя выходного файла:	leftrec.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В теории формальных грамматик и автоматов (ТФГиА) важную роль играют так называемые *контекстно-свободные грамматики* (КС-грамматики). КС-грамматикой будем называть четверку, состоящую из множества N нетерминальных символов, множества T терминальных символов, множества P правил (продукций) и начального символа $S \in N$.

Каждая продукция $p \in P$ имеет форму $A \to \alpha$, где A нетерминальный символ ($A \in N$), а α — строка, состоящая из терминальных и нетерминальных символов. Процесс вывода слова начинается со строки, содержащей только начальный символ S. После этого на каждом шаге один из нетерминальных символов, входящих в текущую строку, заменяется на правую часть одной из продукций, в которой он является левой частью. Если после такой операции получается строка, содержащая только терминальные символы, что процесс вывода заканчивается.

Во многих теоретических задачах удобно рассматривать так называемые *нормальные формы* грамматик. Процесс приведения грамматики к нормальной форме часто начинается с устранения *левой рекурсии*. В этой задаче мы будем рассматривать только ее частный случай, называемый *непосредственной левой рекурсией*. Говорят, что правило вывода $A \to \alpha$ содержит непосредственную левую рекурсию, если первым символом строки α является A.

Задана КС-грамматика. Найти количество правил, содержащих непосредственную левую рекурсию.

Формат входного файла

Первая строка входного файла содержит количество n ($1 \le n \le 1000$) правил в грамматике. Каждая из последующих n строк содержит по одному правилу. Нетерминальные символы обозначаются заглавными буквами латинского алфавита, терминальные — строчными. Левая часть продукции отделяется от правой символами `->`. Правая часть продукции всегда непуста и имеет длину не более 30 символов.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

leftrec.in	leftrec.out
3 S->Sabc S->A A->AA	2

Задача F. Матрица

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

Задана матрица K, содержащая n строк и m столбцов. *Седловой точкой* этой матрицы назовем элемент, который одновременно является минимумом в своей строке и максимумом в своем столбце.

Найдите количество седловых точек заданной матрицы.

Формат входного файла

Первая строка входного файла содержит целые числа n и m ($1 \le n, m \le 750$). Далее следуют n строк по m чисел в каждой. j-ое число i-ой строки равно k_{ij}. Все k_{ij} по модулю не превосходят 1000.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

matrix.in	matrix.out
2 2 0 0 0 0	4
2 2 1 2 3 4	1

Задача G. К коду Грея

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

Рассмотрим циклическую последовательность попарно различных чисел $\{a_0, a_1, \ldots, a_{2^n-1}\}$, $0 \le a_i \le 2^n - 1$. Назовем эту последовательность *кодом Грея*, если для любое a_i отличается от левого соседа a_{i-1} и правого соседа a_{i+1} только в одной цифре в двоичной записи этих чисел. Для a_0 левым соседом считается a_{2^n-1}, а для a_{2^n-1} правым соседом считается a_0.

Вася хочет написать программу — игру-головоломку, которая будет позволять пользователю менять местами два любых числа a_i и a_j. Задача игрока — получить код Грея. Модуль, отвечающий за перестановку чисел, Вася берет на себя. А вот Ваша задача — написать программу, которая будет определять после каждой перестановки — является ли последовательность кодом Грея.

Формат входного файла

В первой строке входного файла содержится число n ($1 \le n \le 16$). В следующей строке через пробел перечислены попарно различные числа a_i ($0 \le i \le 2^n - 1$). В третьей строке записано число m — количество перестановок сделанных пользователем. В следующих m ($1 \le m \le 100000$) строках перечислены числа $(i,\ j)$ ($i \ne j, 0 \le i, j \le 2^n - 1$) — индексы переставляемых элементов.

Формат выходного файла

В выходной файл запишите m строчек — в i-той строке запишите «Yes», если после i-той перестановки последовательность стала кодом Грея и «No» в обратном случае.

Пример

togray.in	togray.out
2 0 1 3 2 2 1 2 2 1	No Yes

Задача H. Треугольная область

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

Одним из субъектов Флатландии является Треугольная область. Как следует из ее названия она имеет форму треугольника, вершины которого находятся в точках с координатами (x_1, y_1), (x_2, y_2), (x_3, y_3). Административный центр этой области находится в точке с координатами (x_c, y_c), которая лежит строго внутри указанного треугольника.

Для оценки транспортного и логистического потенциала области, ее руководству понадобилось узнать расстояние от административного центра области до ее границы. Напишите программу, которая вычислит это расстояние.

Формат входного файла

Первая строка входного файла содержит шесть целых чисел — x_1, y_1, x_2, y_2, x_3, y_3. Вторая строка входного файла содержит два целых числа — x_c и y_c. Все числа во входном файле не превосходят 10000 по абсолютной величине. В заданном треугольнике нет тупых углов.

Формат выходного файла

В выходной файл выведите ответ на задачу с точностью не хуже 10^{-6}.

Примеры

tri.in	tri.out
0 0 1 2 2 0 1 1	0.4472135
0 0 0 3 3 0 1 1	0.7071067