

Задача А. Раскраска таблицы умножения

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

Таблицей умножения назовем таблицу размера n строк на m столбцов, в которой на пересечении i -ой строки и j -ого столбца стоит число $i \cdot j$ (строки и столбцы нумеруются с единицы).

В одной из математических школ было решено провести педагогический эксперимент. Для того, чтобы ученикам было проще запоминать таблицу умножения, некоторые числа в ней будут покрашены в красный, некоторые — в синий, а некоторые — в зеленый цвет (оставшиеся числа будут черными).

Процесс покраски чисел можно условно разбить на четыре этапа. На первом этапе все числа красятся в черный цвет. На втором — все четные числа красятся в красный цвет, на третьем — все числа, делящиеся на 3, красятся в зеленый цвет, на четвертом — все числа, делящиеся на 5, красятся в синий цвет.

Директор школы хочет знать, какое количество картриджей для принтеров необходимо закупить для печати таблиц. Поэтому ему необходима информация о том, сколько чисел какого цвета будет в одной раскрашенной таблице умножения n на m . Напишите программу, решающую задачу подсчета соответствующих количеств.

Формат входного файла

Входной файл содержит два натуральных числа n и m ($1 \leq n, m \leq 1000$).

Формат выходного файла

В первой строке выходного файла выведите количество чисел, покрашенных в красный цвет, во второй — в зеленый, в третьей — в синий, в четвертой — в черный. Следуйте формату, приведенному в примере.

Примеры

<code>coloring.in</code>	<code>coloring.out</code>
10 10	RED : 21 GREEN : 39 BLUE : 36 BLACK : 4
5 2	RED : 5 GREEN : 2 BLUE : 2 BLACK : 1

Задача В. Теория игр

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

Одним из интересных объектов, изучаемых в теории игр, являются так называемые *антагонистические игры двух лиц*. Такие игры характеризуются множеством X стратегий первого игрока, множеством Y стратегий второго игрока и функцией выигрыша $K(x, y) (x \in X, y \in Y)$. Если множества стратегий X и Y конечны, то такую игру принято называть *матричной*, так как функцию выигрыша K в этом случае удобно задавать матрицей.

Рассмотрим матричную игру, в которой $X = \{1, \dots, n\}, Y = \{1, \dots, m\}$. Матрицу выигрышей обозначим символом K . *Нижним значением игры* назовем число $\max_{i=1}^n \min_{j=1}^m K_{ij}$. *Верхним значением игры* назовем число $\min_{j=1}^m \max_{i=1}^n K_{ij}$. Отметим также, что игры, у которых нижнее и верхнее значение совпадают, называются *играми с седловой точкой*.

Задана матрица выигрышей K для некоторой матричной игры. Найдите ее верхнее и нижнее значение.

Формат входного файла

Первая строка входного файла содержит целые числа n и m ($1 \leq n, m \leq 100$). Далее следуют n строк по m числу 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 нижнее и верхнее значения игры.

Примеры

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

Задача С. Фонарь

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

«Одна голова хорошо, а две лучше. Одна лампочка хорошо, а две лучше,» — подумал Миша, и решил собрать фонарик с двумя лампочками. Теперь он хочет узнать, насколько фонарик с двумя лампочками лучше, чем фонарик с одной. Для этого Миша осветил фонариком на стену, и каждая из лампочек осветила на ней круг.

Эффективность фонарика Миша хочет оценить через площадь освещенной части стены. Миша догадался измерить координаты центров освещенных кругов и их радиусы (которые оказались одинаковыми), но что делать дальше он не знает. Напишите программу, которая поможет Мише.

Формат входного файла

Во входном файле записаны целые числа x_1, y_1, x_2, y_2 и r ($1 \leq x_1, y_1, x_2, y_2, r \leq 100$) — координаты центров кругов и их радиус соответственно.

Формат выходного файла

В выходной файл выведите площадь, покрытую двумя кругами с точностью не менее трех знаков после десятичной точки.

Примеры

<code>light.in</code>	<code>light.out</code>
1 2 3 4 2	22.84955592153876

Задача D. Живой Журнал

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

Программист Саша участвует в создание блог-сервиса Живой Журнал. Планируется, что этот сервис будет предоставлять гораздо больше возможностей, чем известный всем LiveJournal. В настоящее же время проблему составляет реализация всех базовых возможностей LiveJournal'a. Одной из таких возможностей является поддержка списков друзей для пользователей.

Заданы: список пользователей, являющихся друзьями данного пользователя, и список пользователей, у которых данный пользователь содержится в списке друзей.

Необходимо получить список друзей данного пользователя (**Friends**), список его взаимных друзей (**Mutual Friends**), и список тех пользователей, у кого данный пользователь содержится в списке друзей, но которые не являются его взаимными друзьями (**Also Friend of**).

Формат входного файла

Первая строка входного файла содержит число n ($0 \leq n \leq 200$) друзей данного пользователя. Последующие n строк содержат каждая по одному имени пользователя, который является другом данного. $(n + 2)$ -ая строка содержит число m ($0 \leq m \leq 200$) пользователей, у которых данный содержится в списке друзей. Далее заданы имена пользователей, у которых данный находится в списке друзей. Эти пользователи заданы в том же формате, что и друзья данного.

Имена пользователей — строки длиной не более 20 символов, содержащие только строчные буквы латинского алфавита и символы тире (“-”). Каждый пользователь указан не более одного раза в каждом из списков.

Формат выходного файла

В выходной файл список друзей данного пользователя (**Friends**), список его взаимных друзей (**Mutual Friends**), и список тех пользователей, у кого данный пользователь содержится в списке друзей, но которые не являются его взаимными друзьями (**Also Friend of**). В каждом списке пользователи должны быть отсортированы по алфавиту. Следуйте формату, приведенному в примерах.

Примеры

jj.in
3 vasya-pupkin bill-hates ivan-ivanov 2 vasya-pupkin destroyer
jj.out
Friends: bill-hates, ivan-ivanov, vasya-pupkin Mutual Friends: vasya-pupkin Also Friend of: destroyer
jj.in
0 0
jj.out
Friends: Mutual Friends: Also Friend of:

Задача Е. Компьютерная сеть

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

Компания «Маша и медведи» является самым крупным интернет-провайдером во всем древнем лесу. Именно по этому, с просьбой подключить их к интернету, обратились N поросят. Поросята живут в домиках, расположенных в различных точках (x_i, y_i) . Ближайшая точка подключения расположена в точке (x_{net}, y_{net}) .

Для того чтобы подключиться к сети всем N поросьятам необходимо:

1. провести провод от точки подключения до домика одного из поросят;
2. от подключенного поросенка провести провода ко всем остальным.

При этом провода могут при необходимости пересекаться.

Поскольку поросята платят деньги в зависимости от длины провода, то они хотят минимизировать суммарную стоимость сети, если единица длины провода стоит c тугриков.

Ваша задача — помочь им в этом.

Формат входного файла

В первой строке входного файла находится число N , $1 \leq N \leq 1000$ и c , $1 \leq c \leq 10^4$. В следующих N строках находятся координаты домов поросят (x_i, y_i) . В последней строке записаны координаты точки соединения. Все координаты целые и не превосходят 1000 по модулю.

Формат выходного файла

Выведите минимальную сумму денег, необходимую для подключения всех поросят, с 5 точными знаками после десятичной точки.

Примеры

net.in	net.out
3 2 0 0 1 0 0 1 -1 0	6.0000000000
3 1 1 1 2 2 3 3 4 4	5.6568542495

Задача F. Проверка на симпатичность

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

Рассмотрим таблицу, содержащую n строк и m столбцов, в каждой клетке которой расположен ноль или единица. Назовем такую таблицу *симпатичной*, если в ней нет ни одного квадрата 2 на 2, заполненного целиком нулями или целиком единицами.

Так, например, таблица 4 на 4, расположенная слева, является симпатичной, а расположенная справа таблица 3 на 3 — не является.

1	0	1	0
1	1	1	0
0	1	0	1
0	0	0	0

0	0	1
0	0	1
1	1	1

Задано несколько таблиц. Необходимо для каждой из них выяснить, является ли она симпатичной.

Формат входного файла

Первая строка входного файла содержит количество t ($1 \leq t \leq 10$) наборов входных данных. Далее следуют описания этих наборов.

Описание каждого набора состоит из строки, содержащей числа n и m ($1 \leq n, m \leq 100$), и n строк, каждая из которых содержит по m чисел, разделенных пробелами. j -ое число в $i+1$ -ой строке описания набора входных данных — элемент a_{ij} соответствующей таблицы.

Гарантируется, что все a_{ij} равны либо нулю, либо единице.

Формат выходного файла

Для каждого набора входных данных выведите единственную строку, содержащую слово YES, если соответствующая таблица является симпатичной, и слово NO — в противном случае.

Примеры

nice.in	nice.out
3	YES
1 1	YES
0	NO
4 4	
1 0 1 0	
1 1 1 0	
0 1 0 1	
0 0 0 0	
3 3	
0 0 1	
0 0 1	
1 1 1	

Задача G. Несложное вычисление

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

Задано натуральное число n . Необходимо перевести его в k -ичную систему счисления и найти разность между произведением и суммой его цифр в этой системе счисления.

Например, пусть $n = 239$, $k = 8$. Тогда представление числа n в восьмеричной системе счисления — 357, а ответ на задачу равен $3 \times 5 \times 7 - (3 + 5 + 7) = 90$.

Формат входного файла

Входной файл содержит два натуральных числа: n и k ($1 \leq n \leq 10^9$, $2 \leq k \leq 10$). Оба этих числа заданы в десятичной системе счисления.

Формат выходного файла

В выходной файл выведите ответ на задачу (в десятичной системе счисления).

Примеры

<code>simple.in</code>	<code>simple.out</code>
239 8	90
1000000000 7	-34

Задача Н. Сверхпростые числа

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

Простым числом будем называть натуральное число, большее единицы и делящееся только на единицу и на само себя. Выпишем все простые числа в порядке возрастания и i -ое в этом порядке число обозначим p_i (число 2 при этом будет иметь номер 1). Так, например, $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_{52} = 239$.

Скажем, что число p_i является *сверхпростым*, если $i = p_k$ для некоторого k . Иными словами, сверхпростое число — это простое число, номер которого в списке простых чисел, упорядоченном по возрастанию, является простым числом.

Дано натуральное число k . Упорядочим все сверхпростые числа по возрастанию. Найдите k -ое сверхпростое число в этом порядке.

Формат входного файла

Входной файл содержит натуральное число k ($1 \leq k \leq 500$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

<code>supprime.in</code>	<code>supprime.out</code>
1	3
2	5
3	11
100	3911