

Задача А. АСМ

Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Андрей популярный писатель-фантаст, он проводит мастер-классы для своих читателей. Наиболее популярным из них является *Alien Communication Masterclass (АСМ)*, на котором он учит как поступать в случае встречи с пришельцем или нахождении инопланетного артефакта.

Одна из лекций посвящена извлечению информации из инопланетных записей. Исследования Андрея базируется на математических формулах пришельцев, которые могут дать некоторые знания об организмах пришельцев (например, мы используем десятичную систему счисления потому что у нас 10 пальцев на верхних конечностях).

Предположим для простоты, что пришельцы используют те же самые цифры, что и мы и таким же образом трактуют сложение, вычитание, умножение, скобки и равенство.

Для своей лекции Андрей хочет найти пример равенства, которое выполняется в системах счисления с основаниями $a_1, a_2, \dots, a_N$, но не выполняется в системах счисления с основаниями $b_1, b_2, \dots, b_M$. Найдите для него пример такой формулы.

Формат входных данных

Первая строка входного файла содержит два целых числа N и M ($1 \leq N, M \leq 8$).

Вторая строка содержит N чисел $a_1, a_2, \dots, a_N$

Третья строка содержит M чисел $b_1, b_2, \dots, b_M$

Все числа a_i и b_i различны и лежат в пределах от 2 до 10.

Формат выходных данных

Вывод должен представлять собой корректное математическое равенство, которое выполняется в системах счисления с основаниями $a_1, a_2, \dots, a_N$ и не выполняется в системах с основаниями $b_1, b_2, \dots, b_M$.

Равенство может содержать цифры от 0 до 9, знак плюс $+$, минус и унарный минус $-$, знак умножения $*$, скобки $()$ и знак равенства $=$. Знак равенства должен быть ровно один.

Все пробельные символы будут проигнорированы при проверке. Количество непробельных символов не должно превышать 10000.

Примеры

Входные данные	Выходные данные
1 2 2 3 9	$(10 - 1) * (10 - 1) + 1 = 10$
2 2 9 10 2 3	$2 + 2 = 4$

Задача В. Вычеркивание двух

Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Задана строка S , состоящая из маленьких букв латинского алфавита. Сколько различных строк можно получить при помощи вычеркивания ровно двух символов из S ?

Формат входных данных

Входной файл содержит строку S , записанную в первой и единственной строке файла. Длина строки S от 2 до 100000 символов включительно. Строка S содержит только маленькие буквы латинского алфавита.

Формат выходных данных

Выходной файл должен содержать одно целое число, равное количеству различных строк, которые можно получить при помощи вычеркивания ровно двух символов из S .

Примеры

Входные данные	Выходные данные
abbccc	5

Задача С. Числа-подстроки

Максимальное время работы на одном тесте: 2 секунды

Максимальный объем используемой памяти: 64 мегабайта

За один шаг к числу X разрешается прибавить или из числа X разрешается вычесть любое положительное число Y , десятичная запись которого является подстрокой десятичной записи числа X . Стоимость такой операции равна сумме цифр числа Y .

Необходимо за минимальную стоимость получить из числа a число b , при этом все промежуточные числа должны быть положительными и не должны превышать n .

Формат входных данных

Входной файл содержит три целых числа: n, a, b ($1 \leq a, b \leq n \leq 5000$).

Формат выходных данных

Если из числа a нельзя получить число b , выведите в выходной файл одно число -1.

Если такая последовательность преобразований существует, в первой строке выходного файла выведите минимальную стоимость требуемого преобразования. Во второй строке выходного файла выведите число k — количество шагов в преобразовании. В последующих k строках выведите сами шаги преобразования по одному в строке. Каждая строка должна иметь вид +число или -число, в зависимости от того, прибавляется или вычитается очередное число.

Примеры

Входные данные	Выходные данные
20 12 18	5 3 -2 +10 -2
100 5 43	29 8 +5 +1 +1 +1 +13 +26 -5 -4
50 5 43	-1

Задача D. Сусанин

Максимальное время работы на одном тесте:

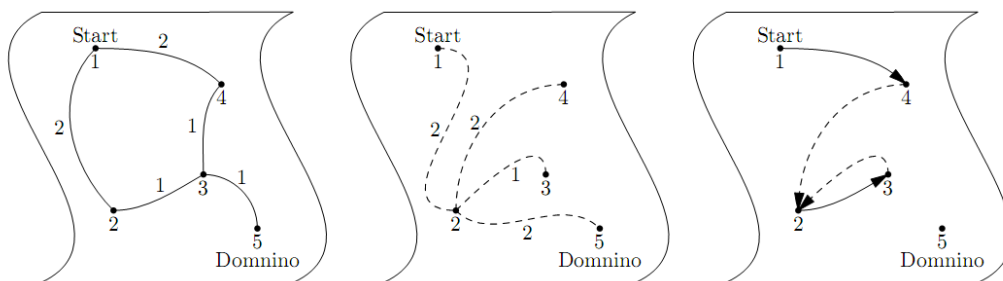
4 секунды

Максимальный объем используемой памяти:

64 мегабайта

Польская армия движется из Костромы в деревню Домнино. Два гетмана Стефан и Константин руководят армией. Стефан изучил карту дорог Костромской губернии и каждую ночь он ведет армию от одной деревни к некоторой другой по некоторой дороге. Константин достал карту секретных троп между деревнями и каждый день он возглавляет марш-бросок армии вдоль одной из этих троп. Каждый гетман перед маршем спрашивает дорогу у Ивана Сусанина.

Длина каждой дороги указана на карте Стефана. Поэтому Стефан знает минимальное расстояние от каждой деревни до Домнино, аналогично Константин знает минимальное расстояние до Домнино по своей карте. Иван Сусанин каждый раз выбирает дорогу для Стефана или тропу для Константина так, что минимальное расстояние между войском и Домнино по соответствующей карте все время строго убывает.



Помогите Ивану найти самый длинный путь в Домнино, удовлетворяющий этим условиям.

Формат входных данных

Первая строка входа содержит три целых числа N , S и T – количество деревень, номер начальной деревни и номер деревни Домнино ($2 \leq N \leq 1000$, $1 \leq s, t \leq N$). Начальная точка не совпадает с Домнино.

Далее идут описание карты Стефана и карты Константина.

Первая строка описания карты содержит число M – количество дорог или троп соответственно ($1 \leq M \leq 100000$). Каждая из следующих M строк содержит 3 целых числа a , b и l – описывающих дорогу/тропу между пунктами a и b с указанием длины l ($1 \leq a, b \leq N$; $1 \leq l \leq 10^6$).

Формат выходных данных

Выведите общую длину пути (вдоль дорог и троп), который Сусанин заставит пройти польскую армию. Если он может заставить армию ходить вечно, так и не достигая Домнино, то выведите -1.

Входные данные	Выходные данные
5 1 5 5 1 2 2 1 4 2 2 3 1 3 4 1 5 3 1 4 1 2 2 2 4 2 2 3 1 2 5 2	-1
3 1 3 4 1 2 10 2 3 10 1 3 20 2 3 30 4 2 1 10 1 3 10 1 1 10 2 3 10	20

Задача Е. Рамки

Максимальное время работы на одном тесте:

2 секунды

Максимальный объем используемой памяти:

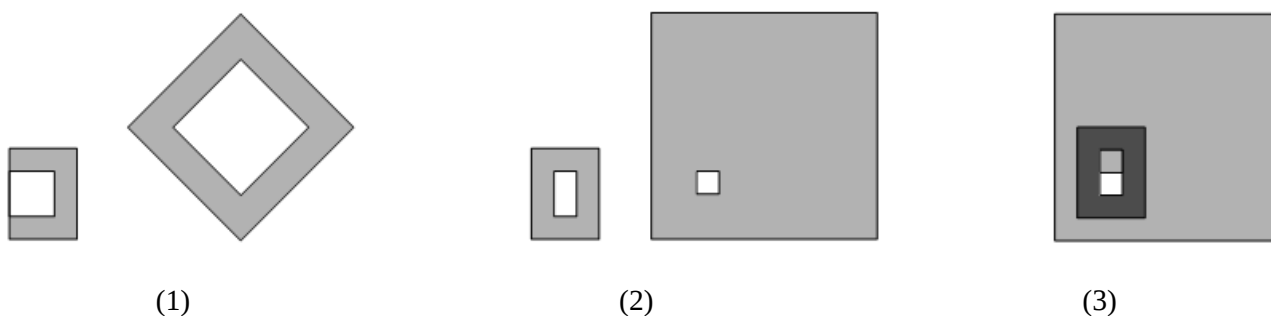
64 мегабайта

Вася с Петей играют в интересную игру. Правила ее очень простые: есть две рамки и участники должны передвинуть вторую рамку так, что площадь пересечения рамок была бы как можно большей. Оба игрока думают в течение минуты и записывают вектор переноса второй рамки. Побеждает тот игрок, чей вектор переноса дал большую площадь пересечения.

Игра имеет много частных случаев, так что Вася решил сжульничать и написать программу, находящую лучший вектор переноса.

В этой игре рамкой называется фигура, состоящая из двух прямоугольников: внешнего и внутреннего. Внутренний прямоугольник лежит строго внутри внешнего. Стороны обоих прямоугольников параллельны координатным осям.

На первом рисунке указаны (1) неправильные рамки, (2) правильные рамки, (3) пересечение рамок.



Площадь рамки вычисляется как $W \cdot H - w \cdot h$, где W, H – размеры внешнего прямоугольника, а w, h – размеры внутреннего ($0 < w < W$; $0 < h < H$).

Напишите программу, которая находит перенос одной рамки относительно другой, которая даст максимальную площадь пересечения.

Формат входных данных

Каждая рамка описывается с помощью четырех точек — двух противоположных углов внешнего прямоугольника и двух противоположных углов внутреннего прямоугольника. Точки описываются своими координатами x и y , которые являются целыми числами и не превосходят 10^8 по модулю.

В первой строке содержится описание первой рамки, во второй — второй.

Формат выходных данных

Первая строка выходного файла должна содержать целое число A – максимальная площадь области пересечения данных рамок достижимая с помощью переноса.

Вторая строка должна содержать пару целых чисел x и y – координаты вектора переноса второй рамки который даст область пересечения A . Координаты не должны превосходить 10^{18} по абсолютному значению.

Пример

Входные данные	Выходные данные
2 2 5 6 3 3 4 5	10
0 0 10 10 2 2 3 3	1 1

Задача F. Гаджеты

Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта

Товарищ Иванов — очень богатый любитель гаджетов. Недавно он понял, что не может купить всех гаджетов, которые он хочет потому что их никто не производит. Поэтому он решил построить свою собственную Фабрику Гаджетов.

Фабрика Гаджетов будет построена на Сколковском шоссе. На этом шоссе сконцентрировано производство нанотехнологичной продукции, необходимой для производства гаджетов. Сколковское шоссе представляет собой прямую, а фабрики — точки на этой прямой (фабрики расположены вдоль дороги).

Для производства гаджета необходимо N нанотехнологичных деталей, M фабрик производит эти детали. Товарищ Иванов хочет минимизировать сумму квадратов расстояний до ближайших фабрик, производящих необходимые детали. Помогите ему и найдите все точки для которых эта сумма минимальна.

Формат входных данных

Первая строка входного файла содержит два натуральных числа N и M ($1 \leq n \leq 10000$; $N \leq M \leq 100000$)

Следующие M строк содержат пары чисел x_i и p_i , x_i — координата i -ой фабрики, а p_i — идентификатор детали, которую производит данная фабрика ($x_i \leq 100000$; $x_i \leq x_{i+1}$; $1 \leq p_i \leq N$).

Для каждой детали существует хотя бы одна фабрика, которая производит эту деталь.

Формат выходных данных

Первая строка выходного файла должна содержать целое число K — количество точек, где может быть расположена Фабрика Гаджетов.

Следующие K строк должны содержать координаты этих точек в возрастающем порядке. Значения должны выводиться с точность 10^{-6} .

Примеры

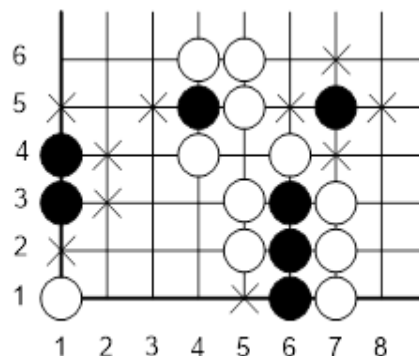
Входные данные	Выходные данные
3 5 -1 3 0 1 2 3 4 2 5 2	1 2.0
2 5 1 1 2 2 3 1 4 2 5 1	4 1.5 2.5 3.5 4.5

Задача G. Го

Максимальное время работы на одном тесте:
Максимальный объем используемой памяти:

1 секунда
64 мегабайта

Совсем недавно Али-Баба узнал от своего брата Касима об удивительной игре Го. В Го играют на прямоугольной доске – гобане, расчерченном вертикальными и горизонтальными линиями. Все линии пронумерованы. В игре участвуют два игрока, которые по очереди выставляют на гобан камни – специальные круглые фишки. Каждый камень ставится на незанятую точку пересечения линий доски (пересечения называют пунктами). У одного игрока – черные камни, у другого – белые. Камни одного цвета, смежные по вертикали, либо по горизонтали (но не диагонали), объединяются в группу. Одиночный камень также считается группой.



Один из способов набрать очки в Го – захватить камни противника. Каждый камень может иметь от двух до четырех смежных с ним пунктов (по вертикали и горизонтали, но не по диагонали). Если такой пункт не занят камнем, то он называется «дамэ». Дамэ группы – это все дамэ камней, составляющих группу. Как только оппонент своими камнями закрывает все дамэ чужой группы, то эта группа считается захваченной и снимается с доски. Если у группы осталось лишь одно дамэ, то говорят, что эта группа находится в «атари» т.е. на один шаг от захвата соперником.

Дамэ черных камней на рисунке отмечены крестиком. Группа черных из камней (1, 3) и (1, 4) имеет 4 дамэ. Группа (6, 1), (6, 2) и (6, 3) имеет одно дамэ и находится в атари. Черный камень (4, 5) также находится в атари. Помогите Али-Бабе, который всегда играет черными, определить, какие его группы находятся в атари.

Формат входных данных

Описание каждого теста начинается со строки, содержащей число N – размерность игровой доски ($6 \leq N \leq 19$). Далее следует N строк по N символов каждая. Каждый символ описывает один пункт доски. «B» означает черный камень, «W» – белый, «.» означает пустой пункт. Все группы на доске имеют хотя бы одно дамэ.

Формат выходных данных

Выводите одно число – количество групп черных камней, находящихся в атари.

Примеры

Входные данные	Выходные данные
9WW.... ...BW.B.. B..W.W... B...WBW..WBW.. W....BW..	2
6 WB.WBB .B.W.B ..WW.W WWW..W ..W... BBW...	1

Задача Н. Борьба с пробками

Максимальное время работы на одном тесте:

2 секунды

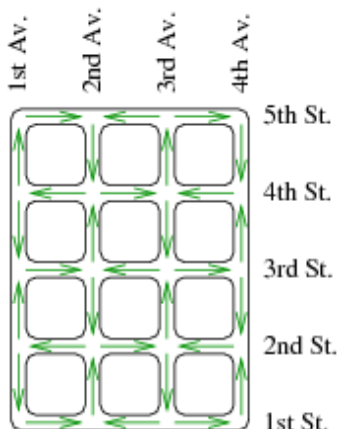
Максимальный объем используемой памяти:

64 мегабайта

Город Нью-Йорк практически парализован гигантским количеством пробок. Мэр Джулио Джулиани предложил новую схему движения по городским улицам. Все улицы по этому плану должны стать односторонними.

Город расположен на прямоугольном острове размером M на N квадратных футов. (M футов с юга на север и N футов с запада на восток. Город разделен на кварталы улицами, идущими с запада на восток и авеню, идущими с юга на север. Длина каждого из кварталов составляет 200 футов (строго говоря, это расстояние между двумя параллельными улицами, окружающими квартал). Улицы нумеруются с юга на север начиная с единицы. Авеню нумеруются с запада на восток также с единицы.

Одностороннее движение организовано так, как показано на рисунке.



Мэр хочет проехать от одного перекрестка до другого. Определите минимальное расстояние между этими перекрестками. Для простоты можно игнорировать расстояние, пройденное на перекрестках, т.е. расстояние, пройденное вдоль квартала составляет ровно 200 футов.

Формат входных данных

Входной файл состоит из 4 строк

Первая строка содержит число M ($400 \leq M \leq 1000000$). Следующая строка содержит число N ($400 \leq N \leq 1000000$). Оба числа делятся на длину квартала. Следующие две строки описывают начальный и конечный перекресток. Каждое описание перекрестка задается так:

m Av., n St.

Например,

1st Av., 5th St.

Формат выходных данных

Выведите минимальное расстояние между перекрестками.

Пример

Входные данные	Выходные данные
800 600 1st Av., 1st St. 2nd Av., 3rd St.	1000

Задача I. Перекраска полосы

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Дана полоска $N \times 1$ клетку, каждая клетка которой раскрашена в один из M цветов. За один ход разрешается перекрасить непрерывную область одного цвета в любой другой цвет.

Требуется определить наименьшее число перекрашиваний, за которое можно получить полосу одного (любого) цвета.

Входные данные

В первой строке находятся два числа N и M – ширина полосы и количество цветов соответственно. $1 \leq N \leq 100$, $1 \leq M \leq 100$. Во второй строке находятся N чисел, соответствующих цветам каждой из клеток полосы от 1 до N (сами цвета лежат в диапазоне от 1 до M , каждый цвет встречается хотя бы один раз).

Выходные данные

Выведите одно число – минимальное число перекрашиваний, за которое можно получить полосу одного цвета.

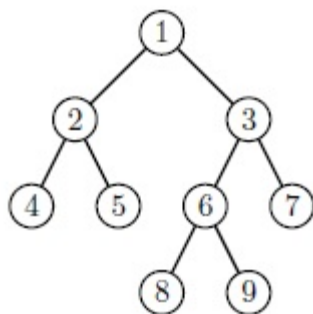
Примеры

Входные данные	Выходные данные
5 3 3 2 1 1 3	2

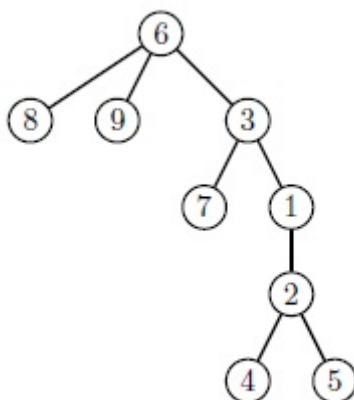
Задача J. Динамическая задача о наименьшем общем предке

Максимальное время работы на одном тесте:	3 секунды
Максимальный объем используемой памяти:	128 мегабайт

Постановка задачи о *наименьшем общем предке* такова: дано дерево T с выделенным корнем и две вершины u и v , $lca(u, v)$ - вершина с максимальной глубиной, которая является предком и u , и v . Например, в картинке внизу $lca(8, 7)$ – вершина 3.



С помощью операции $chroot(u)$ мы можем менять корень дерева, достаточно отметить u , как новый корень, и направить ребра вдоль пути от корня. Наименьшие общие предки вершин поменяются соответственно. Например, если мы сделаем $chroot(6)$ на картинке сверху, $lca(8, 7)$ станет вершина 6. Получившееся дерево изображено снизу.



Вам дано дерево T . Изначально корень этого дерева - вершина 1. Напишите программу, которая поддерживает эти две операции: $lca(u, v)$ и $chroot(u)$.

Входные данные

Входной файл состоит из нескольких тестов. Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строки содержат по 2 натуральных числа и описывают ребра дерева. Далее идет строка с единственным натуральным числом m — число операций —. Следующие m строк содержат операции. Строка "? $u\ v$ " означает операцию $lca(u, v)$, а строка "! u " - $chroot(u)$. Последняя строка содержит число 0.

Сумма n для всех тестов не превосходит 100 000. Сумма m для всех тестов не превосходит 200 000.

Выходные данные

Для каждой операции "? $u\ v$ " выведите значение $lca(u, v)$. Числа разделяйте переводами строк.

Примеры

Входные данные	Выходные данные
9	2
1 2	1
1 3	3
2 4	6
2 5	2
3 6	3
3 7	6
6 8	2
6 9	
10	
? 4 5	
? 5 6	
? 8 7	
! 6	
? 8 7	
? 4 5	
? 4 7	
? 5 9	
! 2	
? 4 3	
0	