

Для всех задач:

Входной файл:	<i>input.txt</i>
Выходной файл:	<i>output.txt</i>
Ограничение по памяти:	64 МБ
Максимальное количество баллов за задачу:	100 баллов

Задача 1. Объём

Ограничение по времени на 1 тест: 2 сек

Даны N параллелепипедов со сторонами, параллельными осям координат. Найдите объём фигуры, получающейся в их объединении.

Входные данные

В первой строке входного файла записано целое число N — количество параллелепипедов ($2 \leq N \leq 200$).

В следующих N строках даны описания параллелепипедов, по одному на строке. Описание каждого параллелепипеда состоит из шести целых чисел — координат двух противоположных его вершин, причем каждая координата первой вершины строго меньше соответствующей ей координаты второй. Все координаты — целые числа, по модулю не превосходящие 500000.

Выходные данные

В выходной файл необходимо вывести объём объединения заданных параллелепипедов.

Примеры

<i>input.txt</i>	<i>output.txt</i>
2 0 0 0 1 1 1 -1 -1 -1 0 0 0	2
2 0 0 0 1 1 1 -1 -1 -1 1 1 1	8

Задача 2. Робот-круг

Ограничение по времени на 1 тест: 2 сек

По плоскости разбросаны точки. По этой же плоскости перемещается робот-круг, который эти точки собирает по заданной программе. Ваша задача — оценить эффективность его программы. Для этого нужно определить, сколько точек соберет робот.

Программа робота состоит из нескольких циклов. Циклы выполняются последовательно: сначала первый цикл выполняется заданное число раз, затем второй, и так далее. Циклы не могут быть вложенными. Цикл может содержать следующие команды:

- +X** — перемещение параллельно первой оси координат в сторону увеличения первой координаты на 1 единицу;
- +Y** — перемещение параллельно второй оси координат в сторону увеличения второй координаты на 1 единицу;
- X** — перемещение параллельно первой оси координат в сторону уменьшения первой координаты на 1 единицу;
- Y** — перемещение параллельно второй оси координат в сторону уменьшения второй

II ЭТАП ВСЕСИБИРСКОЙ ОТКРЫТОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ
19 декабря 2011 года – 31 января 2012 года

координаты на 1 единицу.

Перед выполнением первой команды программы центр робота находится в начале координат — точке (0, 0). Считается, что робот подбирает точку, если в какой-то момент она оказывается внутри круга или на его границе.

Входные данные

В первой строке входного файла записано целое число R — радиус робота ($1 \leq R \leq 50$).

Во второй строке задано целое число N — количество точек, разбросанных по плоскости ($1 \leq N \leq 10000$). В следующих N строках даны координаты точек, по два числа в строке. Обе координаты — целые числа, не превосходящие по модулю 10^6 . Все точки попарно различны.

Далее во входном файле на отдельной строке дано число M — количество циклов в программе робота ($1 \leq M \leq 10$). В следующих M строках содержатся описания циклов, по одному на строке. Описание каждого цикла содержит два целых числа — количество повторений и количество команд в цикле, за которыми приводится список команд. Количество повторений не превышает 1000, количество команд не превышает 10. Команды даны в формате, описанном в условии. Команды разделены пробелами.

Выходные данные

В выходной файл необходимо вывести одно целое число — количество точек, которые соберет робот.

Примеры

<i>input.txt</i>	<i>output.txt</i>
5 3 -5 -5 5 5 10 10 2 2 2 -Y -X 9 2 +Y +X	3
5 3 -5 -5 5 5 10 10 2 2 2 -Y -X 8 2 +Y +X	2

Задача 3. Список

Ограничение по времени на 1 тест: 2 сек, для решений на Java 4 сек

В этой задаче нужно построить список целых чисел, выполнив несколько команд вставки элементов. Список в начальный момент пуст.

Входные данные

В первой строке входного файла записано целое число N — количество команд вставки элементов ($1 \leq N \leq 100000$). В следующих N строках, по одной на строке, даны сами команды. Описание команды состоит из двух целых чисел — числа, которое нужно вставить в список (целое, по модулю не превышающее 10^9), и номера элемента списка, перед которым это число нужно вставить. Если число вставляется в конец списка (после последнего элемента), то номером элемента считается число $n + 1$, где n — номер последнего элемента в списке до этой операции вставки. Элементы списка нумеруются с единицы. Гарантируется, что все команды корректны. Команды нужно выполнять в том порядке, в котором они даны во входном файле.

Выходные данные

В выходной файл необходимо вывести через пробел введенные целые числа в порядке их вхождения в построенный список.

Пример

<i>input.txt</i>	<i>output.txt</i>
6 6 1 7 1 8 1 9 1 0 1 -1 5	0 9 8 7 -1 6

Задача 4. Кратчайший путь

Ограничение по времени на 1 тест: 2 сек

В некоторой провинции между некоторыми населенными пунктами проложены односторонние дороги определенной длины. По таким дорогам можно перемещаться из начального пункта в конечный, но не в противоположную сторону. Однако между двумя пунктами могут быть проложены две односторонние дороги, направленные в противоположные стороны. Между двумя пунктами может быть проложено не более одной дороги в каждом направлении. Дороги, направленные в противоположные стороны, могут иметь различную длину.

Петя поселился в населенном пункте А. Работает же он в населенном пункте В. Определите, сколько существует способов попасть из пункта А в пункт В, проехав при этом наименьшее возможное расстояние.

Входные данные

В первой строке входного файла записано целое число N — количество населенных пунктов ($3 \leq N \leq 1000$). Все пункты пронумерованы числами от 1 до N . Пункт А имеет номер 1, а В — номер 2.

Во второй строке задано целое число M — количество дорог ($1 \leq M \leq 10000$). В следующих M строках даны описания дорог.

Описание каждой дороги записано на отдельной строке и состоит из трех целых чисел X , Y и W , записанных через пробел. Числа X и Y — номера населенных пунктов, которые соединяет данная дорога, X — начальный пункт, Y — конечный. W — длина дороги в километрах ($1 \leq W \leq 10000$).

Выходные данные

В выходной файл необходимо вывести одно целое число — количество различных кратчайших путей из пункта А в пункт В. Так как это число может быть очень большим, ответ нужно выводить по модулю 1000000007.

Пример

<i>input.txt</i>	<i>output.txt</i>
5 10 1 2 7 1 3 2 3 1 3 1 5 1 5 3 1 3 5 2 3 2 5 1 4 2 4 2 5 2 4 5	4

Задача 5. Прямоугольник 2

Ограничение по времени на 1 тест: 4 сек., для решений на Java 5 сек.

Прямоугольник из равных по размеру квадратных клеток разрезали на части так, что линии разреза проходят строго по сторонам клеток. Затем каждую часть закрасили в свой цвет.

Нужно собрать из этих частей исходный прямоугольник.

Входные данные

В первой строке входного файла задана ширина и высота исходного прямоугольника — целые числа от 5 до 50.

Во второй строке задано количество частей — целое число от 2 до 26.

Далее даны описания частей прямоугольника. В первой строке каждого описания даны два целых числа W и H — ширина и высота картинки, описывающей часть прямоугольника, соответственно. Оба числа не меньше 1 и не больше 50.

В каждой из следующих H строк содержится ровно W символов — либо '.', если данная клетка не включается в эту часть, либо заглавная латинская буква, если эта клетка входит в данную часть прямоугольника. Все части прямоугольника связные, то есть из каждой клетки можно попасть в каждую, переходя через общую для них сторону. Каждая часть прямоугольника обозначается определенной буквой. Части прямоугольника не отражены и не повернуты.

Выходные данные

В выходной файл необходимо вывести полученный прямоугольник. Если возможно несколько ответов, выведите любой. Гарантируется, что хотя бы один такой прямоугольник существует.

Замечание

Эта задача в данных ограничениях по времени полным перебором на 100 баллов, скорее всего, не решается. Но так как тесты сделаны с помощью датчика случайных чисел, то, возможно, участники олимпиады смогут придумать такое решение, которое пройдет большинство из них.

Пример

<i>input.txt</i>	<i>output.txt</i>
5 5 5 4 4 . .AA . .AA AAA. A... 5 2 .BBBB BBBB. 2 2 CC CC 1 1 D 2 3 .E .E EE	CCAAE CCAAE AAAE ABBBB BBBBD

Задача 6. ООО "Вектор"

Ограничение по времени на 1 тест: 1 сек.

Молодая, динамично развивающаяся компания "Вектор" открыла свой новый офис в одном из районов города. Офис представляет собой квадратное помещение. Пока в новый офис не провели интернет, решено было воспользоваться открытой wi-fi точкой доступа, находящейся по соседству. Однако, оказалось, что подключение стабильно только на расстоянии, не превосходящем r . Узнав об этом факте, гендиректор компании захотел узнать, какова площадь той части офиса, которая обеспечена интернетом.

Входные данные

В первой строке входного файла содержатся три целых числа — координаты точки доступа x_s, y_s , и радиус её действия r , во второй строке — координаты угла офиса x_q, y_q , и длина его стороны a . Таким образом, другие углы офиса задаются, как $(x_q + a, y_q)$, $(x_q, y_q + a)$, $(x_q + a, y_q + a)$.

Выходные данные

В выходной файл необходимо вывести вещественное число S — площадь обеспеченной интернетом части офиса с точностью, не менее чем 10^{-5} .

Пример

<i>input.txt</i>	<i>output.txt</i>
0 1 4 -3 0 2	4
0 -1 4 -3 0 2	3.93987658

Задача 7. Разбиение множества

Ограничение по времени на 1 тест: 1 сек.

Вам дано множество, содержащее N различных элементов.

Необходимо определить количество способов разбить данное множество на M непустых подмножеств без учёта порядка элементов.

Входные данные

Первая строка входного файла содержит два числа N и M ($1 \leq N, M \leq 10^3$).

Выходные данные

В выходной файл необходимо вывести одно число — ответ на задачу, взятый по модулю 1000000007 (10^9+7).

Пример

<i>input.txt</i>	<i>output.txt</i>
3 2	3

Замечание

Существует три способа разбить множество $\{1, 2, 3\}$ на два подмножества: $\{1\}\{2, 3\}$, $\{2\}\{1, 3\}$ и $\{3\}\{1, 2\}$.

Задача 8. О лифтах

Ограничение по времени на 1 тест: 1 сек.

Рассматриваем N -этажный дом. В нем всего один подъезд и на этажи людей доставляют M лифтов. Все лифты имеют постоянную скорость движения, равную 1 этаж в секунду. Время остановки лифта на этаже равно T секундам. Один лифт не может перевозить одновременно более Q человек. На каждом этаже есть только две кнопки для вызова лифта – одна вверх и одна вниз. На первом и последнем этажах – по одной кнопке: вверх и вниз, соответственно.

Нужно написать программу, определяющую минимальное время ожидания лифта пассажиром, который на этаже F нажал одну из кнопок, причем считать, что других пассажиров, ожидающих лифта, нет. Следует заметить, что если через этаж, на котором стоит человек, проходит лифт, едущий в противоположную сторону, то он данного пассажира брать не будет. Если лифт пустой, он стоит на том этаже, где его покинул последний пассажир.

Входные данные

В первой строке входного файла записаны четыре целых числа через пробел N, M, Q, T ($1 \leq N, M \leq 255$ и $1 \leq Q, T \leq 10$).

Вторая строка содержит информацию о пассажире — это два числа, первое из которых — номер этажа F , а второе — номер нажатой кнопки (1 – вверх, 2 – вниз)

Следующие M строк описывают состояния лифтов. Каждая строка соответствует одному лифту. Первое число здесь — это номер этажа, на котором находится лифт, затем два числа задают его состояние движения. Возможны следующие варианты:

1 1 — лифт идет вверх, 1 2 — идет вниз, 0 К — стоит К секунд.

Далее дается загруженность лифта (количество человек, находящихся в лифте), и, наконец, описывается план остановок: количество следующих остановок — целое число R , за которым следуют R ($0 \leq R$) пар чисел. Первое число в паре — это номер этажа, где лифт должен остановиться, а второе — количество пассажиров, едущих до этого этажа (остановки могут быть неупорядочены). Количество пассажиров, которых надо высадить, всегда больше нуля. Если лифт идет вверх, то среди остановок не может быть этажей, расположенных ниже этажа, на котором находится лифт в данный момент. Аналогично, если лифт направляется вниз, то в списке остановок могут присутствовать только номера этажей, расположенных ниже текущего.

Выходные данные

В выходной файл необходимо записать одно целое число — минимальное время ожидания лифта.

Пример

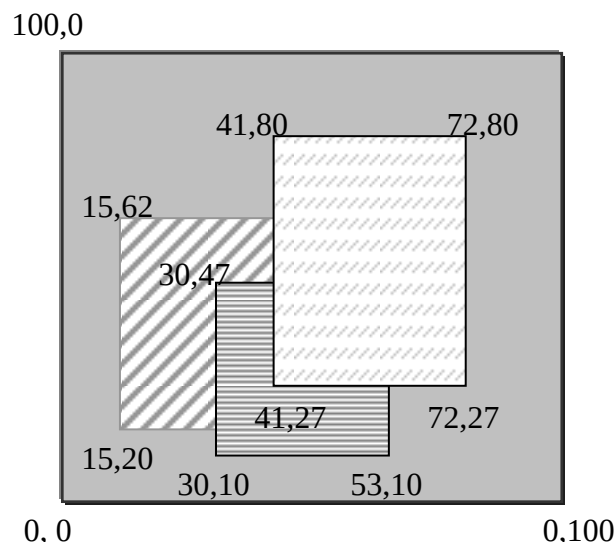
<i>input.txt</i>	<i>output.txt</i>	<i>Комментарии</i>
9 4 5 10 5 2 7 0 2 5 3 4 3 2 1 1 1 4 1 1 2 1 6 2 4 1 2 3 2 3 2 2 1 1 0 0 0	13	9 этажей, 4 лифта, вместимость – 5 человек, время стоянки – 10 секунд Пассажир на 5 этаже нажал кнопку вниз. Лифт на 7 этаже стоит 2 секунды, в нем находится 5 человек, ему нужно сделать 3 остановки: на 4 этаже высадить 3 человек, на 2-м – одного, и на 1-м тоже одного пассажира. Лифт на 4 этаже идет вверх, везет 2 человек, нужно сделать 1 остановку на 6 этаже и высадить этих двоих. Лифт на 4 этаже идет вниз, везет 3 пассажиров, нужно сделать 2 остановки, на 3 этаже высадить двоих человек, на 2-м – одного. Лифт стоит на первом этаже 0 секунд, он пустой, у него нет плана остановок.

Задача 9. О листочках бумаги

Ограничение по времени на 1 тест: 1 сек.

Прямоугольные листы бумаги разложены на поверхности квадратного стола так, чтобы их стороны лежали параллельно краям стола. Листы могут перекрываться. При этом нет листов, которые полностью закрыты другими листами, и нет таких листов, которые касаются сторонами. Введем систему координат, связанную со столом. Левый нижний угол стола становится началом координат. Длина стороны стола равна 100. Координаты принимают целые положительные значения.

Требуется написать программу, которая определяет минимально возможное количество листов бумаги, разложенных на столе, если видно N углов, заданных своими координатами.



Входные данные

В первой строке входного файла записано число N — количество видимых углов листов ($4 \leq N \leq 50$). Каждая следующая строка содержит по два целых числа, записанных через пробел — координаты этих углов. Задача не требует проверки корректности входных данных.

Выходные данные

В выходной файл необходимо вывести одно целое число — количество листов бумаги на столе.

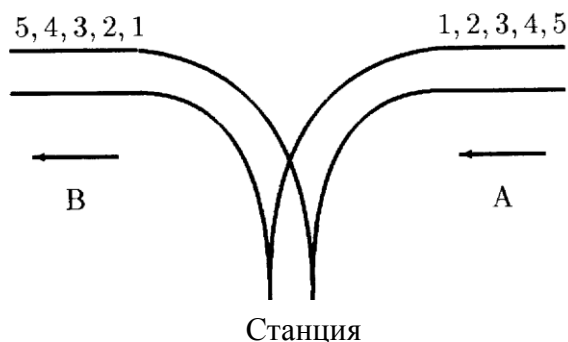
Пример

<i>input.txt</i>	<i>output.txt</i>
9 15 62 30 47 30 10 15 20 41 27 41 80 53 10 72 80 72 27	3

Задача 10. Железная дорога

Ограничение по времени на 1 тест: 1 сек., для решений на Java 2 сек.

В городе ПушПоп есть знаменитая железнодорожная станция. Местность здесь невероятно холмистая. Станция была построена в позапрошлом столетии. К сожалению, денег в то время было очень мало. Было возможно проложить дорогу только на поверхности. Более того, оказалось, что станция может быть только тупиком (смотрите рисунок) и из-за особенностей местности могут быть проложены только одни пути.



Согласно местной традиции каждый поезд, прибывающий из направления А, продолжает свой путь в направлении В с перестроенными некоторым способом вагонами. Предположим, что поезд, прибывающий из А, состоит из N вагонов, пронумерованных в возрастающем порядке $1, 2, \dots, N$. Диспетчер по формированию поездов должен знать, можно ли так перестроить вагоны, чтобы они двигались в сторону В в порядке $a_1, a_2, \dots, a_N$. Помогите ему и напишите программу, которая определит, возможен ли желаемый порядок вагонов или нет. Заметим, что каждый вагон может быть отцеплен от поезда перед тем, как ему прибыть на станцию, и может двигаться сам до тех пор, пока не попадет на пути, ведущие в сторону В. Вы также должны предполагать, что в любое время на станции может скопиться столько вагонов, сколько необходимо. Но если вагон попал на станцию, он не может вернуться в сторону А, и если он покинул станцию в направлении В, он не может возвратиться назад.

Входные данные

Входной файл состоит из нескольких тестовых наборов. Первая строка каждого тестового набора содержит целое число N — количество вагонов ($1 \leq N \leq 1000$).

Следующие строки содержат перестановки элементов множества $\{1, 2, \dots, N\}$.

Последняя строка тестового набора — это 0.

Последняя строка входного файла содержит еще один 0.

Количество строк во входном файле не превосходит 800.

Выходные данные

Каждая строка выходного файла соответствует строке с перестановкой во входном файле. В строку записывается слово **Yes**, если можно перестроить вагоны в порядке, заданном в соответствующей строке входного файла. В противном случае строка должна содержать слово **No**. Последней нулевой строке тестового набора соответствует пустая строка. Последней нулевой строке входного файла нет соответствующей строки в выходном файле.

Пример

<i>input.txt</i>	<i>output.txt</i>
5	Yes
1 2 3 4 5	No
5 4 1 2 3	
0	Yes
6	
6 5 4 3 2 1	
0	
0	