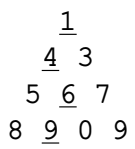
Задача A. Спуск с горы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

В одном из горнолыжных курортов Италии проводятся соревнования по горнолыжному спуску. Каждому спортсмену предстоит скатиться с горы на лыжах. На любом этапе спуска участник получает определённое число очков. После прохождения трассы очки суммируются. Участник, набирающий наибольшее количество очков, выигрывает. Гора представляет собой треугольник, в качестве элементов которого выступают целые числа — очки за прохождение этапа. На каждом уровне спортсмену предоставляется выбор — двигаться вниз влево или вниз вправо. Начало спуска — в самой высокой точке горы, конец — в любой из самых низких.



Требуется найти максимальное количество очков, которое может набрать спортсмен.

Формат входных данных

В первой строке содержится целое число n — количество этапов ($1 \leqslant n \leqslant 100$). Далее следуют n строк, каждая из которых характеризует один уровень. В i-й из этих строк содержится ровно i целых чисел: a_1, a_2, …, a_i — количество очков в каждой из позиций ($-100 \leqslant a_k \leqslant 100$ для всех $1 \leqslant k \leqslant i$).

Формат выходных данных

Выведите одно целое число: максимальное количество очков, которое может набрать спортсмен.

Пример

стандартный ввод	*стандартный вывод*
4 1 4 3 5 6 7 8 9 0 9	20

Задача B. Поле

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Отряду нужно пересечь прямоугольное поле размера $m \times n$ квадратов, двигаясь из левого верхнего угла в правый нижний и перемещаясь между соседними квадратами только в двух направлениях — вправо и вниз. Поле не очень ровное, но у отряда есть карта, на которой отмечена высота каждого квадрата. Опасность перехода с квадрата высоты h_1 на соседний квадрат высоты h_2 оценивается числом $|h_2 - h_1|$; опасность всех переходов в пути суммируется. Выясните, какова минимальная опасность пути из квадрата $(1, 1)$ в квадрат (m, n).

Формат входных данных

В первой строке заданы два числа m и n через пробел ($1 \leqslant m, n \leqslant 100$). В следующих n строках записано по m чисел в каждой; i-е число j-й из этих строк соответствует высоте квадрата (i, j). Все высоты — целые числа в диапазоне от 1 до 100, включительно.

Формат выходных данных

Выведите одно число — минимальную опасность пути из квадрата $(1, 1)$ в квадрат (m, n).

Примеры

стандартный ввод	*стандартный вывод*
2 2 1 1 1 1	0
4 2 1 2 3 5 3 8 4 7	6
2 3 1 2 2 3 3 1	4

Задача C. Грибы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Маша решила навестить свою бабушку. Она взяла с собой две корзинки — одну с пирожками, а другую — пустую, для грибов, которые она хочет собрать по пути.

Для того, чтобы попасть к бабушке, Маше необходимо пройти через лес, который представляет собой прямоугольник размером $m \times n$, в некоторых клетках которого растут деревья, а в некоторых — грибы. Маша выходит из клетки $(1, 1)$ и идёт к бабушке в деревню, расположенную в клетке (m, n). Каждым своим ходом Маша может пойти вправо или вниз (то есть увеличить одну и только одну из своих координат на 1), если в клетке, в которой она после этого окажется, не стоит дерево. Если в обеих клетках и справа, и снизу, находятся деревья, то Маша считается заблудившейся.

Вам необходимо по данному лесу выяснить, может ли Маша дойти до бабушки, не заблудившись, и если может, то посчитать максимальное количество грибов, которое она может при этом собрать.

Формат входных данных

В первой строке находятся четыре числа m, n, g, t ($2 \leqslant m, n \leqslant 100$, $0 \leqslant g, t \leqslant g+t \leqslant m \cdot n-2$). В следующих g строках расположены по два числа в каждой — x и y-координаты i-го гриба. За ними следуют t строк с описаниями деревьев в аналогичном формате. Ни в какой клетке не может расти больше одного гриба, гриб и дерево одновременно, или больше одного дерева. Кроме того, в клетках $(1, 1)$ и (m, n) ничего не растет.

Формат выходных данных

Если Маша может дойти до бабушки, то в первой строке необходимо выдать максимальное количество грибов, которое она сможет при этом собрать, а в последующих $m + n - 1$ строках нужно выдать координаты клеток, последовательно посещаемых Машей, в формате x_i y_i, для пути, на котором достигается максимальное количество грибов. Если таких путей несколько, то разрешается выдавать любой из этих путей.

В противном случае нужно вывести единственное число -1.

Примеры

стандартный ввод	*стандартный вывод*
4 4 3 2 1 4 2 3 4 3 2 2 3 4	2 1 1 1 2 1 3 2 3 3 3 4 3 4 4
2 2 0 2 1 2 2 1	-1

Задача D. Редакционное расстояние

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

В информатике *редакционным расстоянием* между двумя строками называется минимальное количество добавлений, удалений и замен символов, при помощи которых можно из одной строки получить другую. К примеру, редакционное расстояние между строками «`ab`» и «`ab`» равно нулю, так как строки равны между собой безо всяких изменений; расстояние между строками «`short`» и «`ports`» равно трём: в слове «`short`» нужно удалить из начала букву «`s`», заменить «`h`» на «`p`» и добавить в конец букву «`s`». Редакционное расстояние также называют *расстоянием Левенштейна*.

Найдите редакционное расстояние между двумя заданными строками.

Формат входных данных

В первой строчке задана одна строка, во второй — другая. Длины обеих строк — от 1 до 100 символов, включительно.

Формат выходных данных

Выведите единственное число — редакционное расстояние между двумя заданными строками.

Примеры

стандартный ввод	*стандартный вывод*
`ab` `ab`	`0`
`short` `ports`	`3`

Задача E. Наибольшая общая подпоследовательность

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке записано число N — длина первой последовательности ($1 \leqslant N \leqslant 1000$).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие $10\,000$ по модулю.

В третьей строке записано число M — длина второй последовательности ($1 \leqslant M \leqslant 1000$).

В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие $10\,000$ по модулю.

Формат выходных данных

Выведите единственное целое число: длину наибольшей общей подпоследовательности, или же число 0, если такой не существует.

Пример

стандартный ввод	*стандартный вывод*
3 1 2 3 4 2 1 3 5	2

Задача F. Три последовательности

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Даны три последовательности целых чисел. Ваша задача — найти их наибольшую общую подпоследовательность.

Формат входных данных

Входные данные содержат описание трёх последовательностей. Каждая последовательность задаётся в двух строках. Первая из них содержит длину последовательности n ($1 \leqslant n \leqslant 100$), а вторая — её элементы (32-битные целые числа).

Формат выходных данных

Первая строка должна содержать длину максимальной общей подпоследовательности. Саму подпоследовательность необходимо вывести во второй строке. Если таких подпоследовательностей несколько, можно вывести любую из них.

Примеры

стандартный ввод	*стандартный вывод*
3 1 2 3 3 2 1 3 3 1 3 5	2 1 3
3 1 2 3 3 4 5 6 3 1 3 5	0

Задача G. Шаблоны

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и прочего. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имён файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки («.»). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: «?» и «*». Знак вопроса («?») соответствует ровно одному произвольному символу. Звёздочка «*» соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строчке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким образом, как описано выше. Например, строчки «ab», «aab» и «beda.» подходят под шаблон «*a?», а строчки «bebe», «a» и «ba» — нет.

Формат входных данных

Первая заданная строка определяет шаблон P. Вторая строка S состоит только из символов алфавита. Её необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

Формат выходных данных

Если данная строка подходит под шаблон, выведите «YES». Иначе выведите «NO».

Примеры

стандартный ввод	*стандартный вывод*
k?t*n kitten	YES
k?t?n kitten	NO

Задача H. Два шаблона

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и прочего. Ваша задача — найти строку минимально возможной длины, которая подходит под два заданных шаблона.

Алфавит в этой задаче состоит из маленьких букв латинского алфавита и точки («`.`»). Шаблоны могут содержать любые символы алфавита, а также специальные символы «`?`» и «`*`». Под «`?`» подходит любой символ алфавита, а под «`*`» — любая строка символов алфавита (возможно, пустая). Под символы алфавита, встречающиеся в шаблоне, подходят только такие же символы алфавита. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отбразить в строку вышеуказанным способом. Например, строки «`ab`», «`aab`» и «`beda.`» подходят под шаблон «`*a?`», а строки «`bebe`», «`a`» и «`ba`» — нет.

Формат входных данных

Входные данные состоят из одного или нескольких тестовых случаев. В первой строке записано одно целое число — количество тестовых случаев.

Каждый тестовый случай состоит из двух строк, содержащих шаблоны P_1 и P_2. Длина любого из шаблонов не превосходит 100 символов.

Формат выходных данных

Для каждого из тестовых случаев ответ задаётся одной строкой:

- Если строка, подходящая под оба шаблона, существует, выведите такую строку минимально возможной длины (если таких несколько, разрешается выводить любую).
- В противном случае выведите строку «`NO`».

Пример

стандартный ввод	*стандартный вывод*
2 *k*tt*n* *i*e* haha hihi	kitten NO

Задача I. Робот

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Роботу дали задание пройти по лабиринту. Этот лабиринт представляет из себя прямоугольное поле размера $m \times n$ клеток, в котором каждая клетка либо занята сплошной стеной, либо свободна. Каждая клетка описывается парой координат (x, y), где $1 \leqslant x \leqslant m$ и $1 \leqslant y \leqslant n$. Особенности конструкции робота таковы, что он из клетки (x, y) может попасть только в клетки $(x - 1, y)$, $(x + 1, y)$ и $(x, y + 1)$, конечно, если они свободны, но не может уменьшить свою координату по y и перейти на клетку $(x, y - 1)$.

Робот хочет провести в лабиринте как можно больше времени, так как после прохождения лабиринта его наверняка заставят заниматься чем-то более тяжёлым. Однако, если он остановится хоть на секунду или просто замедлит свое движение, могут возникнуть подозрения, что он сломался, и тогда ему грозит попадание на свалку металлолома. Поэтому робот хочет выбрать путь, который бы имел наибольшую длину, и пройти по нему с постоянной скоростью.

Задача робота усложняется тем, что ему нельзя возвращаться на клетку, в которой он уже побывал — иначе баг в программе, которая записывает его путь, заставит его крутиться на одном месте, а о последствиях такой ошибки страшно даже подумать!

Помогите роботу найти самый длинный путь по лабиринту, не имеющий самопересечений и такой, что в нём не уменьшается координата y. Путь может начинаться в любой свободной клетке с $y = 1$ и заканчиваться в любой свободной клетке с $y = n$; можно считать, что весь лабиринт огорожен сплошной стеной, и выходить за его пределы нельзя.

Формат входных данных

В первой строке ввода заданы целые числа m и n через пробел ($1 \leqslant m, n \leqslant 100$). В последующих n строках содержится по m символов в каждой; j-й символ i-й из этих строк равен «`X`» (икс большое), если соответствующая клетка занята стеной, и «`.`» (точка), если она свободна.

Формат выходных данных

Если пути с указаными свойствами не существует, выведите число -1. В противном случае в первую строку выведите число посещённых роботом клеток k, а в последующие k строк по паре целых чисел x_i и y_i через пробел — координаты клеток пути в порядке их посещения. Число k должно быть максимально; если оптимальных ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	***стандартный вывод***
2 2	4 1 1 2 1 2 2 1 2
3 4 .X. ... X.X ..X	6 3 1 3 2 2 2 2 3 2 4 1 4
1 1 X	-1

Задача J. Малое редакционное расстояние

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Рассмотрим две строки S и T, состоящие из сиволов с ASCII-кодами от 33 до 126 включительно. *Редакционное расстояние* между строками — минимальное количество вставок, удалений и замен символов, которые необходимы для того, чтобы преобразовать одну из этих строк в другую. Вам необходимо найти редакционное расстояние между S и T или сообщить, что оно больше заданного числа k.

Формат входных данных

В первой строке задана строка S, во второй — строка T, а в третьей — целое число k ($0 \leqslant |S|, |T| \leqslant 100\,000$, $0 \leqslant k \leqslant 50$).

Формат выходных данных

Если искомое редакционное расстояние больше k, то выведите в первой строке слово «`Infinity`». В противном случае выведите редакционное расстояние.

Примеры

стандартный ввод	*стандартный вывод*
abacabadabacaba abacabadabacaba 24	0
abcdef acdefu 7	2
aaaaaaaaaa bbbbbbbbbb 5	Infinity