Задача A. Магический лабиринт

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

Один лишь ум судья, лишь он способен
Магический осилить Лабиринт...
И там лишь человек увидит слитно,
Что на Земле он видит по частям...

Касиды Хаджи Абду аль-Язди

Вы играете в новейшую компьютерную игру «Heroes of Keyboard and Magic», один из уровней которой представляет собой магический лабиринт. Он состоит из n комнат, которые соединены m коридорами, при этом по каждому коридору можно двигаться в двух направлениях. Ваша задача на этом уровне состоит в том, чтобы добраться от комнаты номер s до комнаты номер t.

Каждый коридор характеризуется цветом — если коридор является белым, то по нему могут перемещать только те персонажи игры, которые являются белыми магами, а если коридор является черным — то только те персонажи, которые являются черными магами. Ваш игровой персонаж является белым магом, поэтому для него может представлять некоторую трудность добраться от одной комнаты до другой.

К счастью, в некоторых комнатах находятся специальные переключатели, которые позволяют менять цвет некоторых коридоров. Такой переключатель выглядит, как кнопка, при нажатии на которую цвет коридоров из определенного набора меняется на противоположный.

Ваша задача состоит в том, чтобы написать программу, которая по информации о лабиринте и по номерам начальной комнаты и комнаты, в которую требуется попасть, определит минимальное число коридоров, которые необходимо пройти.

Формат входного файла

Первая строка входного файла содержит число n ($1 \le n \le 100$) комнат в лабиринте и число m ($0 \le m \le 1000$) коридоров в лабиринте.

Каждая из последующих m строк содержит по три числа u, v и c — номера комнат, соединенных соответствующем коридором ($1 \le u, v \le n$, $u \ne v$), и цвет этого коридора ($0 \le c \le 1$, черному цвету соответствует $c = 0$, белому — $c = 1$). Любые две комнаты соединены не более чем одним коридором. По каждому из коридоров разрешается двигаться в обоих направлениях.

Следующая строка содержит число k переключателей, установленных в лабиринте ($1 \le k \le 10$). Каждая из последующих k строк содержит описание одного из переключателей — это описание состоит из номера r комнаты, в которой этот переключатель установлен ($1 \le r \le n$), числа c коридоров, которые меняют свой цвет на противоположный при использовании этого переключателя, c чисел — номеров этих коридоров (коридоры нумеруются натуральными числами в том порядке, в котором они перечислены во входном файле).

Последняя строка входного файла содержит два целых числа: номер стартовой комнаты s и номер t комнаты, в которую требуется попасть ($1 \le s, t \le n$).

Формат выходного файла

В выходной файл выведите ответ на задачу. Если из комнаты s попасть в комнату t невозможно, выведите в выходной файл `-1`.

Примеры

maze.in	maze.out
3 2 1 2 0 2 3 0 2 1 1 1 2 2 1 2 1 3	2
3 2 1 2 0 2 3 0 2 1 1 1 2 1 1 1 3	-1

В первом примере необходимо воспользоваться сначала переключателем в первой комнате, затем пройти по коридору (он станет белым) во вторую комнату, затем воспользоваться переключателем во второй комнате и перейти по коридору (он также станет белым) в третью комнату.

Во втором примере нет никакой возможности перейти из второй комнаты в третью, поэтому в третью комнату попасть невозможно.

Задача B. Корень

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

> В армии сержант:
> — Так, всем копать! Это кто тут математическую школу окончил? Ты, Сидоров? Так, бери лопату, будешь корни извлекать!
>
> Фольклор

В математике часто рассматриваются так называемые «вычеты по модулю p» — множество чисел от 0 до $p-1$. Наиболее простыми свойствами эти множества обладают в том случае, если p является простым числом. Именно такой случай и рассматривается в этой задаче.

Для вычетов рассматривают операции сложения, вычитания, умножения, деления и некоторые другие. Ваша задача состоит в том, чтобы написать программу для нахождения корня k-ой степени. Для некоторого упрощения задачи корень k-ой степени надо будет искать не для произвольного числа, а для единицы.

Более формально, ваша задача состоит в том, чтобы найти все такие числа x, что $x^k \bmod p = 1$. Искомых корней может быть несколько, например, при $p = 5$, $k = 2$ существуют два корня: 1 ($1^2 \bmod 5 = 1$) и 4 ($4^2 \bmod 5 = 16 \bmod 5 = 1$).

Формат входного файла

Первая строка входного файла содержит два целых числа: p и k ($2 \le p \le 10^9$, p — простое, $2 \le k \le p-1$).

Формат выходного файла

В первой строке выходного файла выведите число n найденных корней, а во второй строке — сами корни в возрастающем порядке. Гарантируется, что число корней не будет превышать 10000.

Примеры

root.in	root.out
5 2	2 1 4
31 12	6 1 5 6 25 26 30

Задача C. Дерево разбора

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

Арифметические выражения, использующие сложение, вычитание, умножение, деление и возведение в степень определяются следующей грамматикой:

$$\begin{aligned}
\langle\text{выражение}\rangle &\longrightarrow \langle\text{слагаемое}\rangle \mid \langle\text{выражение}\rangle + \langle\text{слагаемое}\rangle \mid \langle\text{выражение}\rangle - \langle\text{слагаемое}\rangle \\
\langle\text{слагаемое}\rangle &\longrightarrow \langle\text{множитель}\rangle \mid \langle\text{слагаемое}\rangle * \langle\text{множитель}\rangle \mid \langle\text{слагаемое}\rangle \,/\, \langle\text{множитель}\rangle \\
\langle\text{множитель}\rangle &\longrightarrow \langle\text{элемент}\rangle \mid \langle\text{элемент}\rangle\ \hat{}\ \langle\text{множитель}\rangle \\
\langle\text{элемент}\rangle &\longrightarrow \langle\text{переменная}\rangle \mid (\langle\text{выражение}\rangle) \\
\langle\text{переменная}\rangle &\longrightarrow a \mid b \mid \ldots \mid z
\end{aligned}$$

Сложение, вычитание, умножение и деление левоассоциативны, а возведение в степень правоассоциативно.

Для арифметического выражения определено его *дерево разбора*. Это двоичное дерево, в котором внутренние узлы соответствуют бинарным операциям, а листья соответствуют переменным. Дерево строится рекурсивно.

- Дерево для переменной — это дерево из одной вершины, в которой записана эта переменная.
- Дерево для элемента, являющегося выражением в скобках, — это дерево для самого выражения.
- Дерево для множителя, являющегося элементом, — это дерево для этого элемента. Дерево для множителя вида «элемент e, возведенный в степень "множитель f"» — это дерево, в котором в корне записана операция '`^`', левое поддерево корня — дерево для элемента e, правое поддерево корня — дерево для множителя f.
- Деревья для множителя и слагаемого определяются аналогично, с тем лишь различием, что соответствующие операции лево-ассоциативные.

Вам дано арифметическое выражение, выведите его дерево разбора.

Формат входного файла

Во входном файле содержится корректное арифметическое выражение, состоящее не более чем из 400 символов.

Формат выходного файла

В выходной файл выведите дерево разбора.

Дерево разбора для переменной должно быть размера 1×1 и содержать эту переменную.

Дерево, в корне которого записана операция, с поддеревьями T_1 и T_2, которые имеют размеры $h_1 \times w_1$ и $h_2 \times w_2$ соответственно, должно быть размера $(\max\{h_1, h_2\} + 2) \times (w_1 + w_2 + 5)$.

Подробнее о формате вывода можно узнать, изучив пример выходного файла (см. ниже).

Следует использовать следующие вспомогательные символы: минус '`-`' (код ASCII 45), точка '`.`' (код ASCII 46), вертикальная черта '`|`' (код ASCII 124), квадратные скобки '`[`' и '`]`' (коды ASCII 91 and 93).

Пример

<table>
<tr><th>tree.in</th><th>tree.out</th></tr>
<tr><td><pre>(a+b+c)*(d-a)</pre></td><td><pre>   .----[*]----.
         | |
   .----[+]-.   .-[-]-.
   | | | |
.-[+]-.   c   d   a
| |
a   b</pre></td></tr>
</table>

Задача D. Wikipidia

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

Петя работает над новым проектом онлайн-энциклопедии, которую он назвал “Wikipidia”. Сейчас он занимается написание движка по переводу вики-страниц в веб-страницы. Но у него мало времени, поскольку он параллельно занимается раскруткой проекта, поэтому он нанял вас чтобы написать движок.

Для начала движок должен поддерживать полужирный шрифт и курсив. В вики-страницах полужирный шрифт окружается тремя апострофами («'''»), а курсив — двумя апострофами («''»). В веб-страницах полужирный шрифт окружается парой тегов –, а курсив — парой тегов <i>–</i>. Как в вики, так и в веб-страницах маркеры начала/конца полужирного шрифта и курсива должны корректно сопоставляться друг другу и не быть вложенными. А именно, если мы заменим маркеры начала полужирного шрифта на «(», маркер конца полужирного шрифта на «)», маркер начала курсива на «[» и маркер конца курсива на «]», то должна получиться правильная скобочная последовательность, причем никакая круглая скобка не находится внутри другой пары круглой скобки, и никакая квадратная скобка не находится внутри другой пары квадратных скобок.

В отличии от веб-страниц в вики-страницах иногда непросто понять, как применить форматирование, поскольку последовательности из пяти и более апострофов подряд могут быть проинтерпретированы различными способами. Вам надо написать движок, который будет находить корректную интерпретацию, если она существует.

Формат входного файла

Входной файл содержит одну строку, на которой расположена статья в вики-формате, она содержит только буквы латинского алфавита, знаки пунктуации («,», «.», «-», «?», «!»,), пробелы и апострофы, которые используются для форматирования полужирного шрифта и курсива.

Размер входного файла не превышает 100 000 байт.

Формат выходного файла

Выведите в выходной файл веб-версию статьи. Выходной файл не должен содержать апострофов.

Проверка вашего ответа будет осуществляться следующим образом. Проверяющая программа проверит, что парные теги – и <i>–</i> образуют правильную скобочную последовательность и не вложены. После этого каждый из тегов и будет заменен на три апострофа, а каждый из тегов <i> и </i> — на два апострофа. Результат должен в точности совпадать со входной статьей.

Если перевести входную статью невозможно, выведите «!@#$%».

Примеры

wikipidia.in	wikipidia.out
'''''This is''' sample article.''	<i>This is sample article.</i>
'''This is incorrect sample.	!@#$%
''''Funky sample.''''	<i></i>Funky sample.<i></i>