

Problem A. Дробь

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 mebibytes

При обработке результатов физических экспериментов возникла следующая задача: Дано вещественное число X . Необходимо найти рациональное приближение этого числа с точностью до пяти знаков (это значит, что нужно найти такую дробь $\frac{a}{b}$, что ее вещественное представление совпадает как минимум с пятью знаками после запятой у данного числа).

Input

Единственная строка входного файла содержит вещественное число от 0 до 1, имеющее не более 15 знаков после запятой.

Output

Первая строка выходного файла содержит числитель требуемой дроби, вторая строка — знаменатель требуемой дроби.

Числитель и знаменатель данной дроби не должны превосходить 10^9 . Если решений несколько, выведите любое.

Example

standard input	standard output
0.25	1 4

Problem B. Helicopter

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Программа экспериментальной модели автопилота для беспилотного вертолѐта содержит 2 команды управления высотой: «up» (обозначается символом ‘u’) — команда вертолѐту подняться на 1 метр, и «down» (обозначается символом ‘d’) — команда вертолѐту опуститься на 1 метр. Изначально вертолѐт находится на поверхности воды (высота 0). Если вертолѐту поступит команда, в результате которой вертолѐт должен оказаться под водой, вертолѐт тонет. Если в результате выполнения программы вертолѐт оказывается на высоте, большей 0, он падает в воду, переворачивается и тонет. Программа считается корректной, если после её выполнения вертолѐт не утонул.

Вам задана некая (не обязательно корректная!) программа для автопилота, состоящая из команд управления высотой, и несколько последовательно применяемых запросов, состоящих в замене некоторой команды из программы на противоположную («up» на «down» и наоборот). Для каждого запроса определите, является ли программа, полученная после применения этого запроса, корректной.

Input

В первой строке входного файла задана исходная программа — последовательность из N символов ‘u’ и ‘d’ ($1 \leq N \leq 10^5$). Во второй строке содержится количество запросов Q ($1 \leq Q \leq 10^5$). В последующих Q строках заданы запросы. Каждый запрос представляет собой одно целое число, не превосходящее $N - 1$ — номер в программе команды, которая этим запросом меняется на противоположную. Нумерация команд начинается с нуля.

Output

Для каждого запроса в выходной файл выведите в порядке следования запросов **ТАК**, если получившаяся после запроса программа корректна, и **НИЕ** в противном случае.

Example

standard input	standard output
udud	НИЕ
4	ТАК
1	НИЕ
1	ТАК
2	
2	

Problem C. Keyboard

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

На экспериментальном планшетном компьютере текст можно набирать только с помощью стилуса. За одно нажатие на экран пользователь может сделать одно из двух действий:

- T — набрать одну букву на экранной клавиатуре
- C — выделить какой-то фрагмент уже набранного текста и добавить его в конец текста.

По заданному тексту вычислите, какое наименьшее число нажатий на экран потребуется пользователю, чтобы набрать в точности этот текст.

Input

Первая строка входного файла содержит текст, который предстоит набрать пользователю. Текст состоит из не менее, чем 1, и не более, чем 10^4 строчных латинских букв.

Output

В первую строку выходного файла выведите одно число n — количество нажатий на экран. В следующих n строках выведите действия, которые должен проделать пользователь для того, чтобы набрать заданную строку. Если пользователь должен набрать букву, выведите один символ ‘T’. Если пользователь должен скопировать фрагмент, выведите символ ‘C’ и два числа — номер (начиная с 1) первого и последнего символов соответствующего фрагмента. В случае наличия нескольких последовательностей действий минимальной длины выведите любую из них.

Example

standard input	standard output
хуху	3 T T C 1 2
xxx	3 T C 1 1 C 1 1

Problem D. Knight

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Один эксцентричный лорд решил провести экспериментальное «шахматное дерби» — гонки на шахматных конях по доске.

Для достижения успеха требуется решить следующую задачу: даны два поля на шахматной доске. За какое минимальное число ходов конь, расположенный на одном поле, сможет дойти до другого?

Input

В единственной строке входного файла заданы две строки — координаты полей в стандартной шахматной нотации (строка состоит из двух символов: строчной латинской буквы и цифры, где буква от **a** до **h** обозначает номер вертикали, а цифра от 1 до 8 — номер горизонтали). Первым задано исходное поле.

Output

В выходной файл выведите одно число — минимальное количество ходов коня, за которое можно его перевести с начального поля на конечное, или -1 , если такое действие невозможно.

Example

standard input	standard output
e2 e4	2
a1 a1	0

Problem E. Movie

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Во время премьеры экспериментального 4D-фильма все места в кинотеатре оказались проданы. Однако на балкон зрителей запустили только перед самым началом фильма...

На балконе находится N кресел, занумерованных от 1 до N . Вход находится перед первым креслом. Зрители входят на балкон в случайном порядке и рассаживаются согласно купленным билетам. При этом проход, идущий вдоль балкона, является узким и в нём может поместиться только один человек. Каждый зритель проходит до своего места за нулевое время, но на то, чтобы занять место, подключить 4D-оборудование и освободить проход для следующих за ним зрителей, у него уходит 1 минута. Каждый стоящий в проходе зритель занимает одно место: если из-за того, что некий зритель с билетом на место K настраивает оборудование, зритель с билетом на место $L > K$ не может занять своё место, то он ждёт около места $K - 1$, и (если место уже не занято) в свою очередь блокирует доступ к этому месту.

Например, пусть на балконе 5 мест, а зрители идут в порядке 4, 2, 5, 1, 3. Тогда 4-й и 2-й зрители занимают места, 5-й зритель вынужден ждать 2-го 1 минуту около места 1, то есть 1-й зритель тоже не может занять своё место сразу. Через минуту 5-й и 1-й зрители проходят на место, но 3-й зритель не может занять своё место сразу — он вынужден ждать минуту, пока не подключится 1-й. И только потом он занимает своё место и подключает оборудование. Тем самым рассадка заняла 3 минуты.

По заданному порядку входа зрителей на балкон выясните, какое время потребуется всем зрителям, чтобы занять свои места и подключить оборудование..

Input

В первой строке входного файла задано целое положительное число $N \leq 100$ — количество мест на балконе. Во второй строке задана перестановка из N чисел — порядок, в котором зрители входят на балкон.

Output

Выведите одно число — время, которое потребуется для того, чтобы все зрители заняли свои места и подключили оборудование.

Example

standard input	standard output
5 4 2 5 1 3	3

Problem F. STAGE

Input file: **standard input**
Output file: **standard output**
Time limit: **3 seconds**
Memory limit: **64 Mebibytes**

В экспериментальной системе машинного перевода STAGE-3 языковые конструкции представлены в виде деревьев разбора. Исходная фраза разбивается на несколько семантических блоков, каждый из которых, в свою очередь, может состоять из нескольких семантических блоков и так далее. Каждому семантическому блоку сопоставляется его тип — одна из строчных латинских букв *a, b, c, d, e*.

Будем считать, что дерево разбора *A* вложено в дерево разбора *M*, если существует такое соответствие $f(x)$ между вершинами дерева *A* и некоторыми вершинами дерева *M*, что:

- каждой вершине x дерева *A* соответствует ровно одна вершина $f(x)$ дерева *M*, а каждой вершине дерева *M* соответствует не более одной вершины дерева *A*;
- вершина x принадлежит поддереву дерева *A* с корнем в вершине y тогда и только тогда, когда вершина $f(x)$ принадлежит поддереву дерева *M* с корнем в вершине $f(y)$;
- для всех вершин x тип вершины x совпадает с типом вершины $f(x)$.

В целях дальнейшего развития системы, в частности, требуется реализовать следующую подзадачу: задано дерево разбора *M* и несколько запросов. Один запрос состоит в том, чтобы для некоторого дерева разбора A_i выяснить, вложено ли дерево A_i в дерево *M*. Требуется обработать все запросы в порядке поступления.

Input

В первой строке входного файла задано дерево разбора *M*, состоящее из не менее, чем одной, и не более, чем из 10^4 вершин.

Дерево задаётся следующим способом: для каждого семантического блока, начиная с основной фразы, задан его тип — одна из строчных латинских букв *a, b, c, d, e*, затем, если блок разбивается на несколько семантических блоков — задан список таких блоков. Список заключён в круглые скобки, а описания блоков в списке разделены запятой. Так, описание $e(e, d(b, b), a)$ обозначает, что основная фраза имеет тип *e* и разбивается на три блока — первый из них имеет тип *e* и не содержит других семантических блоков, второй имеет тип *d* и разбивается на два семантических блока, каждый из которых имеет тип *b* и не содержит других семантических блоков, третий имеет тип *a* и не содержит других семантических блоков.

Во второй строке входного файла указано количество запросов n ($1 \leq n \leq 100$).

В каждой из последующих n строк входного файла задан один запрос — дерево разбора A_i , содержащее не менее одной и не более 16 вершин.

Output

Для i -го запроса в отдельной строке выведите 1, если дерево A_i вложено в дерево *M*, и 0 в противном случае.

Example

standard input	standard output
$e(e, d(b, b), a)$	1
2	0
$e(b, a)$	
$e(d, b)$	