Задача A. Анти-Фибоначчи

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

Лёше надоели числа Фибоначчи. Всю последнюю неделю, когда он приходил на урок математики или информатики, учителя рассказывали что-то про числа Фибоначчи и задавали на дом задачки про них.

На этой неделе домашнее задание у Лёши — написать программу, которая по заданному целому положительному числу N находит количество способов разбить N на положительные целые слагаемые. Способы, отличающиеся лишь порядком слагаемых, считаются одинаковыми. К примеру, для $N = 4$ это количество способов — 5:

$$\begin{aligned} N &= 4 \\ &= 3+1 \\ &= 2+2 \\ &= 2+1+1 \\ &= 1+1+1+1 \end{aligned}$$

Поскольку Лёше не нравятся числа Фибоначчи, он решил написать программу, которая считает только такие разбиения, в которых среди слагаемых нет чисел Фибоначчи. Более того, разбиения, в которых количество слагаемых является числом Фибоначчи, Лёша тоже решил не считать.

Помогите Лёше написать такую программу.

Формат входного файла

В первой строке входного файла записано целое число N ($1 \leqslant N \leqslant 50$).

Формат выходного файла

Выведите в выходной файл одно число — количество таких разбиений N на положительные целые слагаемые, что ни какое-либо из слагаемых, ни их количество не являются числами Фибоначчи.

Пример

antifib.in	antifib.out
4	0

Задача B. Ход слоном

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

Лёша хочет научиться хорошо играть в шахматы. Для этого он решил написать программу, которая будет играть с ним. Лёша надеется, что потом, улучшая свою программу, он, глядишь, научится играть сам.

В данный момент Лёша работает над модулем, который по заданной позиции определяет, как могут походить какие фигуры. Для начала он решил научить фигуры ходить по пустой доске, а затем уже добавить проверку того, что ход возможен при учёте остальных фигур, стоящих на доске.

Фигура, над которой Лёша работает — слон. Помогите ему по данной позиции шахматного слона определить, в какие клетки шахматной доски он может походить.

Формат входного файла

В первой строке входного файла задана позиция шахматного слона в виде двух символов: первый — номер вертикали (от '`a`' до '`h`'), а второй — номер горизонтали (от '`1`' до '`8`').

Формат выходного файла

В выходной файл выведите столько строк, сколько существует клеток, в которые шахматный слон может походить из данной позиции. В каждой из этих строк выведите координаты какой-то из этих клеток в том же формате, что и во входном файле. Каждая клетка, на которую можно походить, должна встречаться в выходном файле ровно один раз. Порядок вывода клеток не важен.

Пример

bishop.in	bishop.out
a2	b3 c4 d5 e6 f7 g8 b1

Задача C. Максимальная соответствующая подпоследовательность

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

На прошлом занятии кружка программистов Лёше рассказали, как решать задачу о максимальной общей подпоследовательности двух последовательностей. Задача состоит в следующем. Даны две последовательности $a_1 a_2 \ldots a_m$ и $b_1 b_2 \ldots b_n$; пусть для простоты они состоят из маленьких букв латинского алфавита. *Общей подпоследовательностью* этих последовательностей называется такая последовательность $c_1 c_2 \ldots c_k$, что существуют такие индексы $i_1 < i_2 < \ldots < i_k$, что $a_{i_1} = c_1$, $a_{i_2} = c_2$, $\ldots$, $a_{i_k} = c_k$, и такие индексы $j_1 < j_2 < \ldots < j_k$, что $b_{j_1} = c_1$, $b_{j_2} = c_2$, $\ldots$, $b_{j_k} = c_k$. Задача состоит в том, чтобы найти такую последовательность, у которой k было бы максимально.

В качестве домашнего задания было дано следующее обобщение этой задачи:

Пусть задан граф соответствий на буквах латинского алфавита, то есть для некоторых упорядоченных пар букв (α, β) известно, что α *соответствует* β (обратите внимание, что это ещё не значит, что β соответствует α). *Соответствующей подпоследовательностью* назовём такую последовательность $c_1 c_2 \ldots c_k$, что существуют такие индексы $i_1 < i_2 < \ldots < i_k$, что $a_{i_1} = c_1$, $a_{i_2} = c_2$, $\ldots$, $a_{i_k} = c_k$, и такие индексы $j_1 < j_2 < \ldots < j_k$, что c_1 соответствует b_{j_1}, c_2 соответствует b_{j_2}, $\ldots$, c_k соответствует b_{j_k}. Как и прежде, нужно найти такую последовательность, у которой k было бы максимально.

Помогите Лёше справиться с домашним заданием!

Формат входного файла

В первой строке входного файла записаны через пробел три целых числа m, n и l ($1 \leqslant m, n \leqslant 100$, $0 \leqslant l \leqslant 676$). Во второй строке задана последовательность $a_1 a_2 \ldots a_m$ маленьких латинских букв без пробелов. В третьей строке задана последовательность $b_1 b_2 \ldots b_n$ маленьких латинских букв без пробелов. В следующих l строках содержится по две буквы α_r и β_r без пробелов в каждой; эти строки означают, что α_r соответствует β_r.

Формат выходного файла

В первой строке выведите k — максимальную длину соответствующей последовательности. Во второй строке выведите индексы i_1, i_2, $\ldots$, i_k через пробел, а в третьей — индексы j_1, j_2, $\ldots$, j_k также через пробел. Если ответов с максимальным k несколько, разрешается выводить любой из них.

Примеры

lcs.in	lcs.out
1 1 0 a a	0
1 1 1 a b ab	1 1 1
3 5 4 abc cbdfe ac ad bd cf	3 1 2 3 1 3 4