

In this problem, you are given two words x and y , and a finite sequence of words $\{w_1, w_2, \dots, w_k\}$. If it is possible to obtain the same word by appending to x and y some words from the given sequence of words, we say that x and y can be *adjusted*. We would like to check whether the words x and y can be adjusted using the words from the given sequence.

Given a word w , we can perform an operation $w * w_i$, $1 \leq i \leq k$, consisting in appending the word w_i to the word w at the right. We define this as an append operation. The task is to find the smallest number of append operations that are necessary to adjust two given words using the words from a given sequence.

For example, words **abba** and **ab** can be adjusted by the words from the sequence $\{ \text{baaabad}, \text{aa}, \text{badccaa}, \text{cc} \}$. It suffices to append to **abba** two words: **aa** and **badccaa**, and to **ab** three words: **baaabad**, **cc** and **aa**. In both cases we obtain: **abbaaabadccaa**.



Input

The first line of input contains the number of cases that follow. The first two lines of data for each case contain the words x and y , respectively. The third line contains the integer k , $0 \leq k \leq 1000$, which is the length of the sequence of words that can be used for word adjustment. The following k lines contain one word each. All words use only lowercase letters and contain between 1 and 1000 characters.

Output

For each case output one nonnegative integer giving the minimal number of operations that are needed to adjust two given words, or output '-1' if it is impossible.

Sample Input

```
2
abba
ab
4
baaabad
aa
badccaa
cc
a
ab
4
bb
ab
ba
aa
```

Sample Output

```
5
-1
```