Stringing Strings

Max. score: 100

This problem is no longer available for practice. Apology for any inconvenience!

Sam just got a new present from his dad and it's a puzzle game. The puzzle consists of a long string *S* containing *N* lowercase alphabet letters. Beside that, there are *K* different short strings *T* numbered from *1* to *K*. The goal is to recreate long string *S* from the short strings and each short string can only be used once. He doesn't necessarily need to use all K strings to recreate the long string.

Sam thinks it's easy and he can solve it, but his Dad said that he needs to use a minimum number of small pieces to solve the puzzle in the correct manner.

Now Sam is confused, can you help him to solve this puzzle?

It is guaranteed that there is at least 1 configuration for *T* that match with *S*

Input Format

- The first line will be given an integer *N*, which indicates the length of the string
- The second line will be given a string *S*, which indicates the long string
- The third line will be given an integer *K*, which indicates the number of small pieces string
- The next K line will be given a string $T_{i\ ..\ K}$ to indicate the small pieces string numbered from *1* to *K* respectively

Output Format

- The first line of output is an integer *X* which indicates the number of small pieces used to solve the puzzle. In the next line, print *X* integer indicates the sequence number of small pieces used in the correct order starting from the front. It is guaranteed there will be at least one possible correct combination. If there are multiple solutions, print the combination with the smallest lexicographic order.

Constraints

$0 \leq N \leq 2e5$
$1 \leq K \leq 20$
$1 \leq |T_i| \leq 2e5$
$|T_1| + |T_2| + |T_3| + \ldots + |T_K| \leq 5e6$

SAMPLE INPUT

```
5
abcab
3
abc
cab
ab
```

SAMPLE OUTPUT

```
2
1 3
```

Explanation

{3, 2} is also a correct combination since *"ab" + "cab" = "abcab"* but it has larger lexicographic order

Time Limit:	2.0 sec(s) for each input file.
Memory Limit:	512 MB
Source Limit:	1024 KB

Marking Scheme:	Score is assigned when all the testcases pass.
Allowed Languages:	Bash, C, C++, C++14, C++17, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, Java 14, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Python 3.8, Racket, Ruby, Rust, Scala, Swift-4.1, Swift, TypeScript, Visual Basic