Problem F. Fraud Busters

Input file: `fraud.in`
Output file: `fraud.out`

The number of cars in Default City that travel to the city center daily vastly exceeds the number of available parking spots. The City Council had decided to introduce parking fees to combat the problem of overspill parking on the city streets. Parking fees are enforced using an automated vehicle registration plate scanners that take a picture of the vehicle registration plate, recognize the sequence of digits and letters in the code on the plate, and check the code against a vehicle registration database to ensure that parking fees are dutifully paid or to automatically issue a fine to the vehicle owner otherwise.

As soon as parking fees were introduced, a parking fee fraud had appeared. Some vehicle owners had started to close one or several digits or letters on their vehicle registration plate with pieces of paper while they park, thus making it impossible for the current version of the automated scanner to recognize their vehicle's registration code and to issue them a fine.

The Default City Council had instituted the Fraud Busters Initiative (FBI) to design a solution to prevent this kind of fraud. The overall approach that FBI had selected is to expand the number of vehicle features that scanners recognize (including features like vehicle type and color), as well as excluding from the list any vehicles that are detected to be elsewhere at this time. This information should help to identify the correct vehicle by narrowing down the search in the vehicle registration database.

You are working for FBI. Your colleagues had already written all the complex pieces of the recognition software that analyses various vehicle features and provides you with a list of registration codes that might potentially belong to a scanned car. Your task it to take this list and a recognized code from the license plate (which may be partially unrecognized) and find all the registration codes that match.

Input

The first line of the input file contains 9 characters of the code as recognized by the scanner. Code that was recognized by the the scanner is represented as a sequence of 9 digits, uppercase English letters, and characters "`*`" (star). Star represents a digit or a letter that scanner could not recognize.

The second line of the input file contains a single integer number n ($1 \le n \le 1000$) — the number of vehicle registration codes from the vehicle registration database.

The following n lines contain the corresponding registration codes, one code per line. Vehicle registration codes are represented as a sequence of 9 digits and uppercase English letters. All codes on these n lines of the input file are different.

Output

On the first line of the output file write a single integer k ($0 \le k \le n$) — the number of codes from the input file that match the code that was recognized by the scanner. The code from the scanner matches the code from the database if the characters on all the corresponding positions in the codes are equal or the character from the scanner code is "`*`".

On the following k lines write the matching codes, one code per line, in the same order as they are given in the input file.

Sample input and output

fraud.in	fraud.out
A**1MP19* 4 A001MP199 E885EE098 A111MP199 KT7351TTB	2 A001MP199 A111MP199