Problem C
Cyber Guardian

In the good old days, the Internet was free from fears and terrorism. People did not have to worry about any cyber criminals or mad computer scientists. Today, however, you are facing atrocious crackers wherever you are, unless being disconnected. You have to protect yourselves against their attacks.

Counting upon your excellent talent for software construction and strong sense of justice, you are invited to work as a cyber guardian. Your ultimate mission is to create a perfect firewall system that can completely shut out any intruders invading networks and protect children from harmful information exposed on the Net. However, it is extremely difficult and none have ever achieved it before. As the first step, instead, you are now requested to write a software simulator under much simpler assumptions.

In general, a firewall system works at the entrance of a local network of an organization (e.g., a company or a university) and enforces its local administrative policy. It receives both inbound and outbound packets (note: data transmitted on the Net are divided into small segments called packets) and carefully inspects them one by one whether or not each of them is legal. The definition of the legality may vary from site to site or depend upon the local administrative policy of an organization. Your simulator should accept data representing not only received packets but also the local administrative policy.

For simplicity in this problem we assume that each network packet consists of three fields: its source address, destination address, and message body. The source address specifies the computer or appliance that transmits the packet and the destination address specifies the computer or appliance to which the packet is transmitted. An address in your simulator is represented as eight digits such as 03214567 or 31415926, instead of using the standard notation of IP addresses such as 192.168.1.1. Administrative policy is described in filtering rules, each of which specifies some collection of source-destination address pairs and defines those packets with the specified address pairs either legal or illegal.

Input

The input consists of several data sets, each of which represents filtering rules and received packets in the following format:

$n\ m$
$rule_1$
$rule_2$

...

$rule_n$
$packet_1$
$packet_2$

...

$packet_m$

The first line consists of two non-negative integers n and m. If both n and m are zeros, this means the end of input. Otherwise, n lines, each representing a filtering rule, and m lines, each representing an arriving packet, follow in this order. You may assume that n and m are less than or equal to 1,024.

Each $rule_i$ is in one of the following formats:

permit *source-pattern destination-pattern*

deny *source-pattern destination-pattern*

A *source-pattern* or *destination-pattern* is a character string of length eight, where each character is either a digit ('0' to '9') or a wildcard character '?'. For instance, "1????5??" matches any address whose first and fifth digits are '1' and '5', respectively. In general, a wildcard character matches any single digit while a digit matches only itself.

With the keywords "permit" and "deny", filtering rules specify legal and illegal packets, respectively. That is, if the source and destination addresses of a packed are matched with *source-pattern* and *destination-pattern*, respectively, it is *permitted* to pass the firewall or the request is *denied* according to the keyword. Note that a permit rule and a deny rule can contradict since they may share the same source and destination address pair. For the purpose of conflict resolution, we define a priority rule: $rule_i$ has a higher priority over $rule_j$ if and only if $i > j$. For completeness, we define the default rule: any packet is illegal unless being explicitly specified legal by some given rule.

A packet is in the following format:

source-address destination-address message-body

Each of the first two is a character string of length eight that consists solely of digits. The last one is a character string consisting solely of alphanumeric characters ('a' to 'z', 'A' to 'Z', and '0' to '9'). Neither whitespaces nor special characters can occur in a message body. You may assume that it is not empty and that its length is at most 50.

You may also assume that there is exactly one space character between any two adjacent fields in an input line representing a rule or a packet.

Output

For each data set, print the number of legal packets in the first line, followed by all legal packets in the same order as they occur in the data set. Each packet must be written exactly in one line. If the data set includes two packets consisting of the same source and destination addresses and the same message body, you should consider them different packets and so they must be written in different lines. Any extra whitespaces or extra empty lines must not be written.

Sample Input

```
2 5
permit 192168?? ?12??34?
deny 19216899 01234375
19216711 11233340 HiIamACracker
19216891 01234345 Hello
19216899 01234345 HiIamAlsoACracker
```

```
19216809 11200340 World
00000000 99999999 TheEndOfTheWorld
1 2
permit 12345678 23456789
19216891 01234345 Hello
12345678 23456789 Hello
0 0
```

Output for the Sample Input

```
2
19216891 01234345 Hello
19216809 11200340 World
1
12345678 23456789 Hello
```

First Input Data

Your first input data is here.

The ACM ICPC