B. Radio Station

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

As the guys fried the radio station facilities, the school principal gave them tasks as a punishment. Dustin's task was to add comments to nginx configuration for school's website. The school has n servers. Each server has a name and an ip (names aren't necessarily unique, but ips are). Dustin knows the ip and name of each server. For simplicity, we'll assume that an nginx command is of form "`command ip;`" where `command` is a string consisting of English lowercase letter only, and `ip` is the ip of one of school servers.

Each ip is of form "`a.b.c.d`" where a, b, c and d are non-negative integers less than or equal to 255 (with no leading zeros). The nginx configuration file Dustin has to add comments to has m commands. Nobody ever memorizes the ips of servers, so to understand the configuration better, Dustin has to comment the name of server that the ip belongs to at the end of each line (after each command). More formally, if a line is "`command ip;`" Dustin has to replace it with "`command ip; #name`" where `name` is the name of the server with ip equal to `ip`.

Dustin doesn't know anything about nginx, so he panicked again and his friends asked you to do his task for him.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 1000$).

The next n lines contain the names and ips of the servers. Each line contains a string `name`, name of the server and a string `ip`, ip of the server, separated by space ($1 \le |name| \le 10$, $name$ only consists of English lowercase letters). It is guaranteed that all ip are distinct.

The next m lines contain the commands in the configuration file. Each line is of form "`command ip;`" ($1 \le |command| \le 10$, `command` only consists of English lowercase letters). It is guaranteed that ip belongs to one of the n school servers.

Output

Print m lines, the commands in the configuration file after Dustin did his task.

Examples

input

```
2 2
main 192.168.0.2
replica 192.168.0.1
block 192.168.0.1;
proxy 192.168.0.2;
```

output

```
block 192.168.0.1; #replica
proxy 192.168.0.2; #main
```

input

```
3 5
google 8.8.8.8
codeforces 212.193.33.27
server 138.197.64.57
redirect 138.197.64.57;
block 8.8.8.8;
cf 212.193.33.27;
unblock 8.8.8.8;
check 138.197.64.57;
```

output

```
redirect 138.197.64.57; #server
block 8.8.8.8; #google
cf 212.193.33.27; #codeforces
unblock 8.8.8.8; #google
check 138.197.64.57; #server
```