

1、在搜索引擎后端服务中，需要对恶意的抓取进行限制，其中的一个环节即对访问 IP 进行限制。请设计一个 IP 过滤器，实现对访问的 IP 限制的功能。对 IP 的限制数据有三种格式：

1. 全 IP：如 222.205.58.16

2. 前面带 *：如 *.58.16

3. 后面带 *：如 222.205.58.*

带 * 的表示匹配到任意的 IP 段均可，且 * 可以代表多个 ip 段，且 * 只能出现在开头或者结尾。

现给出若干条需要过滤的规则，以及若干输入的 IP，你需要输出这若干条 IP 是否会被过滤

解析：/**

* 前缀 后缀 的匹配问题 easy

*/

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int N = sc.nextInt();
```

```
        int M = sc.nextInt();
```

```
        sc.nextLine();
```

```
        //读取规则
```

```
        String[] patterns = new String[N];
```

```
        for(int i = 0 ; i < N; i++){
```

```
            patterns[i] = sc.nextLine();
```

```
        }
```

```
        //读取 IP
```

```
        String[] IPs = new String[M];
```

```
        for(int i = 0; i < M; i++){
```

```
            IPs[i] = sc.nextLine();
```

```
        }
```

```
        //暴力匹配
```

```
        for(int i = 0; i < IPs.length; i++){
```

```
            boolean lock = false;
```

```
            for(int j = 0; j < patterns.length; j++){
```

```
                String t = "";
```

```
                if(patterns[j].charAt(0) == '*') {
```

```

        t = patterns[j].replace("*", "");
        if (IPs[i].endsWith(t)) {
            System.out.print(l + " ");
            lock = true;
            break;
        }
    } else if (patterns[j].charAt(patterns[j].length() - 1) ==
        '*') {

        t = patterns[j].replace("*", "");
        if (IPs[i].startsWith(t)) {
            System.out.print(l + " ");
            lock = true;
            break;
        }
    } else {

        if (patterns[j].equals(IPs[i])) {
            System.out.print(l + " ");
            lock = true;
            break;
        }
    }
}

if (lock == false) {
    System.out.print(0 + " ");
}

}

}

```