题目描述：

企业路由器的统计页面，有一个功能需要动态统计公司访问最多的网页 URL top N。请设计一个算法，可以高效动态统计 Top N 的页面。

输入描述：

每一行都是一个 URL 或一个数字，如果是 URL，代表一段时间内的网页访问；如果是一个数字 N，代表本次需要输出的 Top N 个 URL。

输入约束：1、总访问网页数量小于 5000 个，单网页访问次数小于 65535 次；2、网页 URL 仅由字母，数字和点分隔符组成，且长度小于等于 127 字节；3、数字是正整数，小于等于 10 且小于当前总访问网页数；

输出描述：

每行输入要对应一行输出，输出按访问次数排序的前 N 个 URL，用逗号分隔。

输出要求：1、每次输出要统计之前所有输入，不仅是本次输入；2、如果有访问次数相等的 URL，按 URL 的字符串字典序升序排列，输出排序靠前的 URL；

示例 1

输入：

```
news.qq.com

news.sina.com.cn

news.qq.com

news.qq.com

game.163.com
```

```
game.163.com

www.huawei.com

www.cctv.com

3

www.huawei.com

www.cctv.com

www.huawei.com

www.cctv.com

www.huawei.com

www.cctv.com

www.huawei.com

www.cctv.com

www.huawei.com

3
```

输出：

```
news.qq.com,game.163.com,news.sina.com.cn

www.huawei.com,www.cctv.com,news.qq.com
```

说明：

示例 2

输入：

```
news.qq.com
www.cctv.com
1
www.huawei.com
www.huawei.com
2
```

```
3
```

输出：

```
news.qq.com
www.huawei.com,news.qq.com
www.huawei.com,news.qq.com,www.cctv.com
```

```
import java.util.*;

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        Map<String, Integer> countMap = new HashMap<String, Integer>();
        PriorityQueue<String> priorityQueue =
        new PriorityQueue<String>((o1, o2) -> {
            if (!countMap.get(o2).equals(countMap.get(o1))) {
                return countMap.get(o2) - countMap.get(o1);
            }
            return o1.compareTo(o2);
        });
        while (in.hasNext()) { // 注意 while 处理多个 case
            String temp = in.next();
            //判断是网址还是次数
            StringBuilder sb = new StringBuilder();
            List<String> polls = new ArrayList<>();
            if (!temp.contains(".")) {
                int i = Integer.parseInt(temp);
                while (i-- > 0) {
                    String poll = priorityQueue.poll();
                    polls.add(poll);
                    sb.append(poll);
                    if (i > 0) {
                        sb.append(",");
                    }
                }
                System.out.println(sb);
                for (String str : polls) {
                    priorityQueue.offer(str);
                }
            } else {
                Integer count = countMap.get(temp);
                if (count == null) {
                    countMap.put(temp, 1);
                    priorityQueue.offer(temp);
```

```
                } else {
                    countMap.put(temp, count + 1);
                    priorityQueue.remove(temp);
                    priorityQueue.offer(temp);
                }
            }

        }
    }
}
```