

JS-评论转换输出-在一个博客网站上

在一个博客网站上，每篇博客都有评论。每一条评论都是一个非空英文字母字符串。

评论具有树状结构，除了根评论外，每个评论都有一个父评论。

当评论保存时，使用以下格式：

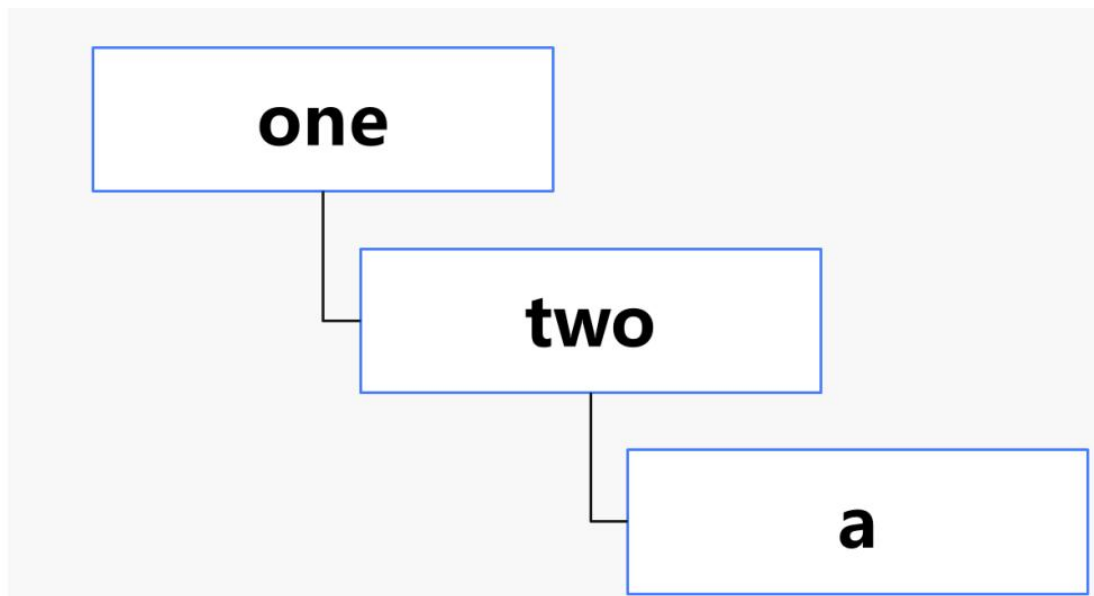
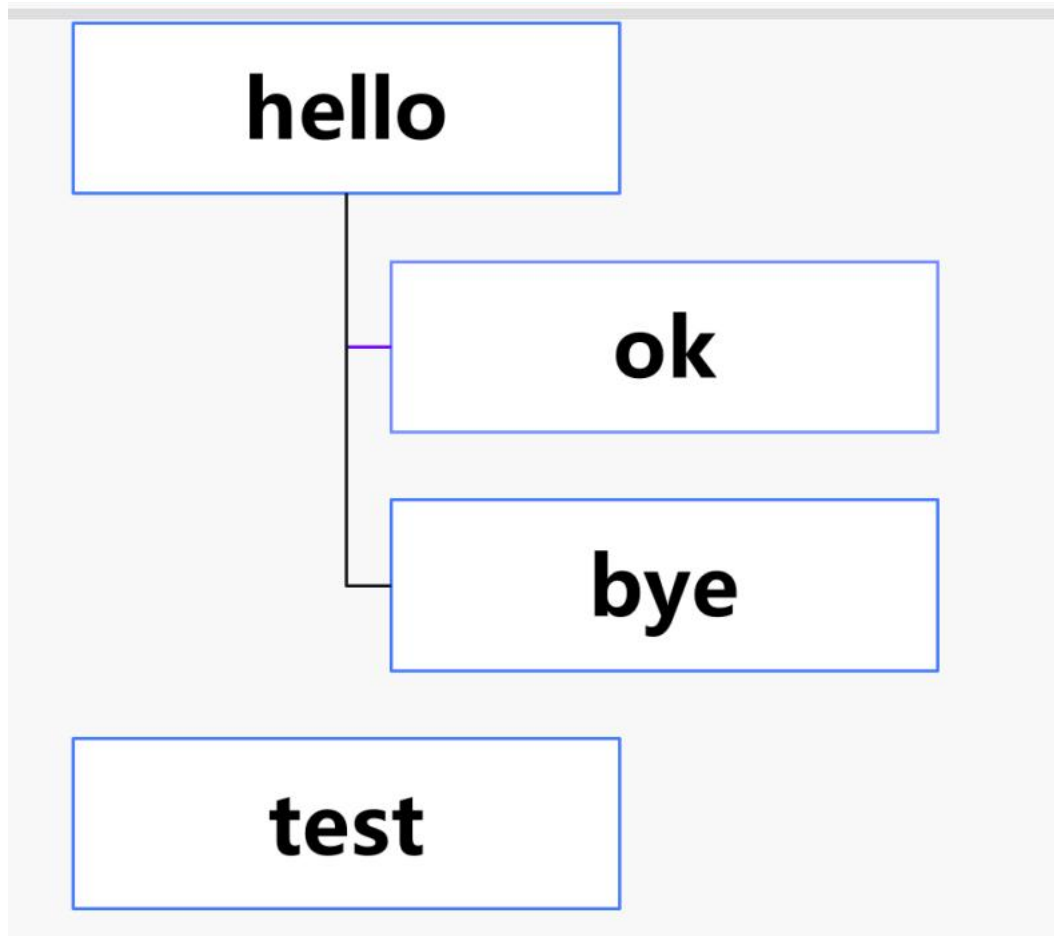
首先是评论的内容；

然后是回复当前评论的数量。

最后是当前评论的所有子评论。（子评论使用相同的格式嵌套存储）

所有元素之间都用单个逗号分隔。

例如，如果评论如下



第一条评论是"hello,2,ok,0,bye,0"，第二条评论是"test,0"，第三条评论是"one,1,two,1,a,0"。

所有评论被保存成"hello,2,ok,0,bye,0,test,0,one,1,two,1,a,0"。

对于上述格式的评论，请以另外一种格式打印：

首先打印评论嵌套的最大深度。

然后是打印 n 行，第 i ($1 \leq i \leq n$) 行对应于嵌套级别为 i 的评论（根评论的嵌套级别为 1）。

对于第 i 行，嵌套级别为 i 的评论按照它们出现的顺序打印，用空格分隔开。

输入描述：一行评论。由英文字母、数字和英文逗号组成。保证每个评论都是由英文字符组成的非空字符串。

每个评论的数量都是整数（至少由一个数字组成）。

整个字符串的长度不超过 10 。

给定的评论结构保证是合法的。

输出描述：按照给定的格式打印评论。对于每一级嵌套，评论应该按照输入中的顺序打印。

补充说明：

示例1

输入：hello,2,ok,0,bye,0,test,0,one,1,two,1,a,0

输出：3

hello test one

ok bye two

a

说明：如题目描述中图所示，最大嵌套级别为3。嵌套级别为1的评论是"hello test one"，嵌套级别为2的评论是"ok bye two"，嵌套级别为3的评论为"a"。

示例2

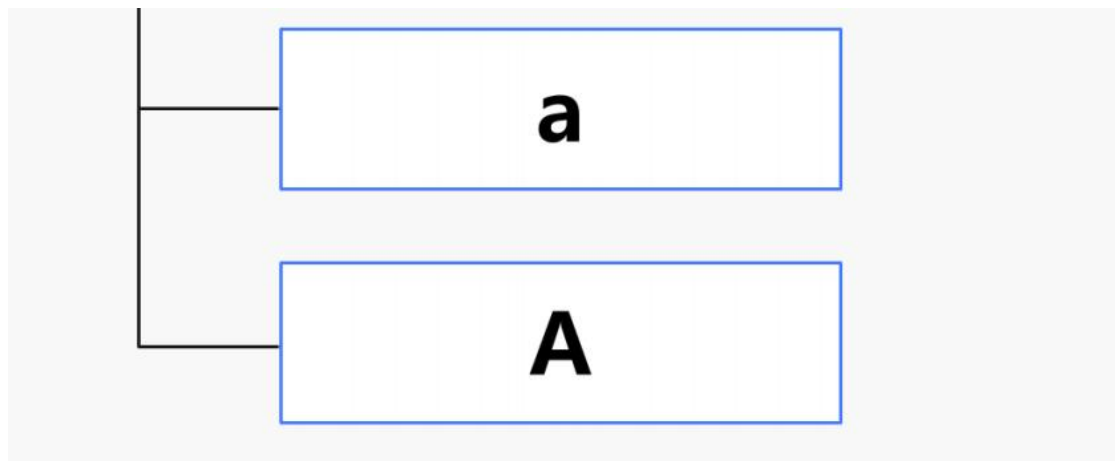
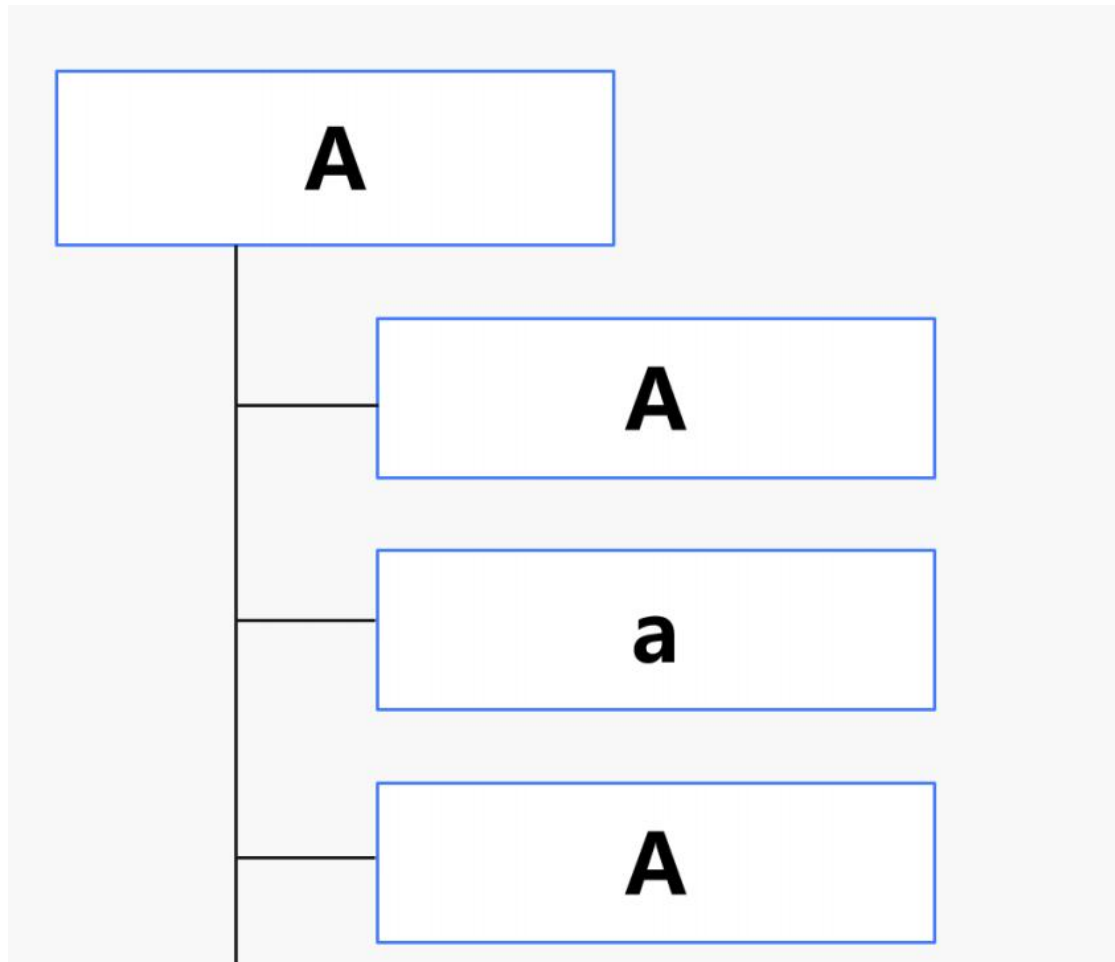
输入：A,5,A,0,a,0,A,0,a,0,A,0

输出：2

A

A a A a A

说明：如下图所示，最大嵌套级别为2，嵌套级别为1的评论是"A"，嵌套级别为2的评论是"A a A a A"



示例3

输入：A,3,B,2,C,0,D,1,E,0,F,1,G,0,H,1,I,1,J,0,K,1,L,0,M,2,N,0,O,1,P,0

输出：4

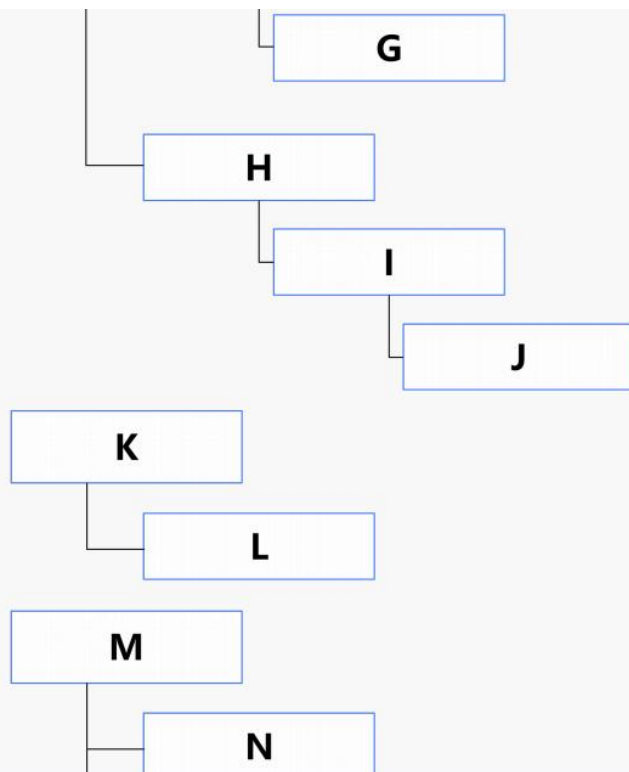
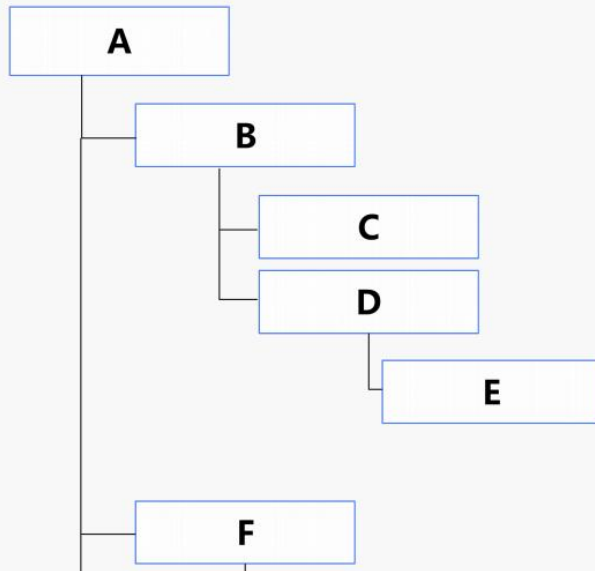
A KM

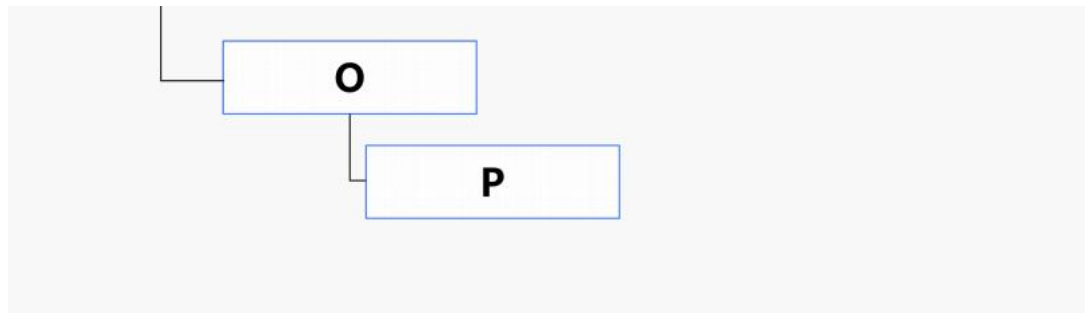
B F H L N O

C D G I P

E J

说明：如下图所示。





```
1  const rl = require("readline").createInterface({ input: process.stdin });
2  var iter = rl[Symbol.asyncIterator]();
3  const readline = async () => (await iter.next()).value;
4
5  void async function () {
6    line = await readline()
7    let tokens = line.split(',');
8    let tokensLength=tokens.length
9    let returnArray=[]
10   let index=0
11   let max=0
12   let calc=function(count){
13     index+=2
14     count++
15     if(returnArray[count-1]==undefined){
16       returnArray.push([])
17     }
18     returnArray[count-1].push(tokens[index-2])
19     if(tokens[index-1]!==0){
20       let number=parseInt(tokens[index-1])
21       for(let i=0;i<number;i++){
22         calc(count)
23       }
24     }else{
25       max=max>count?max:count
26     }
27   }
28   while(index<tokensLength){
29     calc(0)
30   }
31   console.log(max)
32   for(let i=0;i<returnArray.length;i++){
33     console.log(returnArray[i].join(' '))
34   }
35 }()
```