

Problem I. Flat

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

扁平化概念的核心意义是：去除冗余、厚重和繁杂的装饰效果，让"信息"本身重新作为核心被凸显出来。而"扁平化"管理是相对于"等级式"管理构架的一种管理模式。它较好的解决了等级式管理的"层次重叠、冗员多、组织机构运转效率低下"等弊端，加快了信息流的速率，提高决策效率。

对于分为 n 种职级的"等级式"管理的等级结构，每一个职级都有唯一的直系上级，而一个职级可能有若干个(可以0个)下级。即可以将它视为一棵有根树，将职级编号为 $1 \sim n$ ，假设根节点为 u 。

树结构在存储时常常会被扁平化处理为能够表示层次信息的字符串(例如json)，现给出经过扁平化处理的树，请你恢复出原有的树结构。

树是由 n 个节点与 $n - 1$ 条边组成的结构，且树上任意两个节点之间有且仅有一条路径。

核心伪代码如下：

```
def transform(u):
    print(u)
    for v as every son of u:
        print('(')
        transform(v)
        print('')
```

以树根 u 作为参数，执行 $transform(u)$ ，即可得到该树对应的字符串，见样例解释。

Input

第一行为输入数据组数 $T(1 \leq T \leq 10^5)$

后续每两行为一组输入数据

第一行为节点数 $n(1 \leq n \leq 9)$ ，节点编号为 $1 \sim n$

第二行为经过扁平化处理的树，保证是由一棵合法的树经过上述流程转化得到

Output

对于每一组数据，输出一行 n 个数，第 i 个为表示编号为 i 的节点的父节点，假设根节点的父节点为0

Sample

standard input	standard output
3	2 0 2
3	3 4 0 3
2(1)(3)	0 1 2 3 4 5 6 7 8
4	
3(4(2))(1)	
9	
1(2(3(4(5(6(7(8(9))))))))))	

Note

下图为样例2所对应的树结构

