

1. 有  $n$  个同学围成一圈，其  $id$  依次为  $1, 2, 3...n$  ( $n$  号挨着  $1$  号)。现在从  $1$  号开始报数，第一回合报到  $m$  的人就出局，第二回合从出局的下一个开始报数，报到  $m^2$  的同学出局。以此类推直到最后一个回合报到  $m^{(n-1)}$  的人出局，直到剩下最后一个同学。输出这个同学的编号。 $n \leq 15, m \leq 5$

输入:

每一行第一个数字代表  $n$ ，第二个数字代表  $m$

输出:

输出最后剩下同学的编号

示例输入:

5 2

示例输出:

5

2. 弹幕是现今网络视频常见的评论方式，能够反映一个视频的火爆程度。假设某个时间一共有  $N$  条弹幕，每条弹幕  $i$  的持续时间为两个整数表示的时间区间  $[a[i], b[i]]$ ，我们定义弹幕数量最多的一个时间段为最精彩时段，求一个视频的最精彩时段。

输入:

第一行整数  $N$ , 代表弹幕的条数，其中 90% 的  $N < 1000000$ ，60% 的  $N < 10000$   
第二行到第  $N+1$  行，是两个整数  $(a[i], b[i])$ ，代表每条弹幕的开始时间和结束时间，请注意  $(a[i], b[i])$  是全开区间，并且  $a[i], b[i] < 100$

输出:

$M$  行，每行两个整数  $(c, d)$ ， $M$  是答案个数， $(c, d)$  代表视频最精彩时段的开始时间和结束时间，并且  $M$  个答案区间互不重叠。答案请按照开始时间从小到大输出。请注意每行结尾应包含换行符，包括最后一行。

示例输入 1:

3  
0 4  
1 4  
2 3

示例输出 1:

2 3

示例输入 2:

4  
1 2  
3 4  
2 3  
4 5

示例输出 2:

1 2  
2 3  
3 4  
4 5

示例输入 3:

3  
0 2  
2 4  
1 3

示例输出 3:

1 2  
2 3

3. 根据输入生成一棵树，并打印该树。输入为  $N$  行数字对序列（例如： $N = 2$ ，数字对序列为  $(1, 2), (2, 3)$ ），其中数字代表该树的节点，左边数字代表的节点是右边数字代表的节点的父节点，请根据输入的数字对序列生成一棵树，并按照广度优先的顺序打印该树。注意，存在输入无法生成树的情况，比如  $(1, 2) (1, 3) (2, 4) (3, 4)$ ，根据该序列生成的 Graph 中，2 节点和 3 节点同时为 4 节点的父节点，所以根据定义该 Graph 不是树。如遇无法生成树的情况，请输出字符串：“Not a tree”（引号不包括，大小写敏感）。广度优先遍历树并打印输出时，同一层级的节点根据输入时节点出现顺序打印输出。

输入:

输入第一行  $N$  表示数字对序列行树。之后的  $2 \sim N+1$  行表示数字序列对， $1 \leq N \leq 10000$ 。其中，每一行有两个不相同的数字，用英文逗号分隔，例如： $1, 2$ 。所有数字都是正整数，并且小于 2 的 31 次方。没有默认输入顺序，第一行可能不是根节点。最后一行可能不是叶子节点。可能有完全重复的行。

输出:

输出为一行，如果可以生成树，请根据广度优先顺序，输出每个节点对应的数字，并且用英文逗号隔开，同一层级的节点根据输入顺序输出。如果无法生成树，请输出字符串：“Not a tree”（引号不包括，大小写敏感）。生成树中同一层级的节点，按照节点数字在输入时出现的顺序输出。

输入示例 1:

5  
1,2  
1,3  
2,4  
2,5  
3,6

输出示例 1:

1,2,3,4,5,6

输入示例 2:

4  
1,2  
1,3  
2,4  
3,4

输出示例 2:

Not a tree

输入示例 3:

5  
3,6  
2,4  
1,3  
2,5  
1,2

输出示例 3:

1,3,2,6,4,5