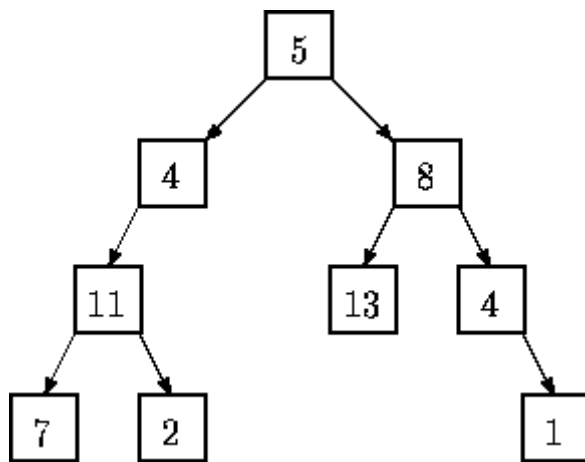


## Trees on the level

樹狀結構在電腦科學的許多領域中都相當重要。本問題牽涉到建立樹及走訪樹。

給你一二元樹，你的任務是寫一個程式來列印依「階層（level-order）」走訪的結果。在本問題中，二元樹的每個節點含有一個正整數，並且節點的數目最少 1 個，最多 256 個。

在「階層」走訪中，依階層從低到高，同階層從左到右的次序來列印。例如以下的二元樹階層走訪的結果為：5, 4, 8, 11, 13, 4, 7, 2, 1



在本問題中，二元樹以節點來表示。每個節點以一對(n,s)來表示。n 代表此節點的值，s 為一字串，代表從根節點到達此節點的路徑。其中 L 代表往左，R 代表往右。所以在上方的圖中內容為 13 的節點其表示法為(13,RL)，而內容為 2 的節點其表示法為(2,LLR)，而根節點為(5,)。

### Input

輸入含有多組測試資料。每組測試資料為若干節點的集合。各節點間以 white space（包含空白、換列等字元）分隔。注意：在各節點內（也就是左刮號到又刮號之間）不會有 white space。當遇到一()的節點，代表該組測試資料結束。請參考 Sample Input。

### Output

對每一組測試資料，如果輸入的節點可以正常的構成一二元樹的話，請輸出按

「階層」走訪的結果。如果輸入的節點無法正常的構成一二元樹的話，也就是說有某些該有的節點沒有給，或重複給（同一路徑有 2 個節點），請輸出 **not complete**。請參考 **Sample Output**。

#### Sample Input

3

(11,LL) (7,LLL) (8,R) (5,) (4,L) (13,RL) (2,LLR) (1,RRR) (4,RR) ()

(3,L) (4,R) ()

(11,LL) (7,LLL) (2,LLL) (8,R) (5,) (4,L) (13,RL) (2,LLR) (1,RRR) (4,RR) ()

#### Sample Output

5 4 8 11 13 4 7 2 1

not complete

not complete

**HINT:**簡單的說就是根據讀入的 **INPUT** 建 **TREE**，如果無法根據讀入的資料建出完整的 **TREE** 則輸出 **not complete**，如果可以建出完整的 **TREE** 則將該 **TREE** 以 **level-order** 的方式走訪並輸出節點上的值