Generate Parentheses

Question: Given n pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

For example, given n = 3, a solution set is:

"((()))", "(()())", "(())()", "()(())", "()()()"

Solutions:

```
class Solution:
    # @param an integer
    # @return a list of string
    def helpler(self, l, r, item, res):
        if r < l:
            return
        if l == 0 and r == 0:
            res.append(item)
        if l > 0:
            self.helpler(l - 1, r, item + '(', res)
        if r > 0:
            self.helpler(l, r - 1, item + ')', res)
    def generateParenthesis(self, n):
        if n == 0:
            return []
        res = []
        self.helpler(n, n, '', res)
        return res
Solution().generateParenthesis(3)
```