class Solution {

public:

vector<string> generateParenthesis(int n) {

vector<string>res;

string path = "";

DFS(res, n, 0, 0, path);

return res;

}

void DFS(vector<string>& res, int n, int k, int left, string& path){

if(left > n) return;

if(k == n){

if(left == 0) res.push\_back(path);

return;

}

path.push\_back('(');

DFS(res, n, k, left + 1, path);

path.pop\_back();

if(left != 0){

path.push\_back(')');

DFS(res, n, k + 1, left - 1, path);

path.pop\_back();

}

}

};