

```

/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
#define ALLOC_LENGTH          (100)

char** result;
int result_index;
int alloc_length;

void _generateParenthesis(int pos, int l, int r, int n)
{
    static char mark[256];
    char* tmp;

    if(pos == 2*n)
    {
        tmp = (char*)malloc((sizeof(char)*2*n)+1);
        memcpy(tmp, mark, (sizeof(char)*2*n));
        tmp[pos] = '\0';
        result[result_index] = tmp;
        result_index++;

        if( (alloc_length % ALLOC_LENGTH) == 0 )
        {
            alloc_length += ALLOC_LENGTH;
            result = (char**)realloc(result, sizeof(char*)*alloc_length);
        }
    }else
    {
        if(l > r)
        {
            mark[pos] = ')';
            _generateParenthesis(pos+1, l, r+1, n);
        }

        if(l < n)
        {
            mark[pos] = '(';
            _generateParenthesis(pos+1, l+1, r, n);
        }
    }
}

char ** generateParenthesis(int n, int* returnSize){

    alloc_length = ALLOC_LENGTH;
    result_index = 0;
    result = (char**)malloc(sizeof(char*)*alloc_length);

    _generateParenthesis(0,0,0,n);
    *returnSize = result_index;

    return result;
}

```