# 1298 Reverse Substrings Between Each Pair of Parentheses (link)

## Description

You are given a string `s` that consists of lower case English letters and brackets.

Reverse the strings in each pair of matching parentheses, starting from the innermost one.

Your result should **not** contain any brackets.

### Example 1:

```
Input: s = "(abcd)"
Output: "dcba"
```

### Example 2:

```
Input: s = "(u(love)i)"
Output: "iloveu"
Explanation: The substring "love" is reversed first, then the whole string is reverse
```

### Example 3:

```
Input: s = "(ed(et(oc))el)"
Output: "leetcode"
Explanation: First, we reverse the substring "oc", then "etco", and finally, the whol
```

### Constraints:

- `1 <= s.length <= 2000`
- `s` only contains lower case English characters and parentheses.
- It is guaranteed that all parentheses are balanced.

(scroll down for solution)

# Solution

*Language: cpp*

**Status: Accepted**

```cpp
#include <string>
#include <stack>
#include <algorithm>

using namespace std;

class Solution {
public:
    string reverseParentheses(string s) {
        stack<string> st;
        string current;

        for (char c : s) {
            if (c == '(') {
                // Когда встречаем открывающую скобку, сохраняем текущую строку в сте
                st.push(current);
                current.clear();
            } else if (c == ')') {
                // Когда встречаем закрывающую скобку, переворачиваем текущую строку 
                reverse(current.begin(), current.end());
                current = st.top() + current;
                st.pop();
            } else {
                // Если это обычный символ, добавляем его к текущей строке
                current += c;
            }
        }

        return current;
    }
};
```