

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

856. Score of Parentheses

Medium

Given a balanced parentheses string s , return *the score of the string*.

The **score** of a balanced parentheses string is based on the following rule:

- $()$ has score 1 .
- AB has score $A + B$, where A and B are balanced parentheses strings.
- (A) has score $2 * A$, where A is a balanced parentheses string.

Example 1:

Input: $s = "()"$

Output: 1

Example 2:

Input: $s = "(())"$

Output: 2

Example 3:

Input: $s = "(())"$

Output: 2

C

```
1 int scoreOfParentheses(char* s) {
2     int score = 0;
3     int depth = 0;
4
5
6     for (int i=0; s[i] != '\0'; i++){
7         if(s[i] == '(') {
8             depth++;
9         } else {
10            depth--;
11            if (s[i - 1] == '('){
12                score += 1 << depth;
13            }
14        }
15    }
16
17    return score;
18 }
```

Saved to local

Ln 11, Col 30

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

$s =$
 $"()"$

Output



Problem List



Run



Submit



0



Premium

Description | Editorial | Solutions | Submissions

856. Score of Parentheses

Medium

Topics

Companies

Given a balanced parentheses string s , return *the score of the string*.

The **score** of a balanced parentheses string is based on the following rule:

- $()$ has score 1 .
- AB has score $A + B$, where A and B are balanced parentheses strings.
- (A) has score $2 * A$, where A is a balanced parentheses string.

Example 1:

Input: $s = "()"$

Output: 1

Example 2:

Input: $s = "(())"$

Output: 2

Example 3:

Input: $s = "()()"$

Output: 2

5.3K



6



</> Code

C v Auto



```
1 int scoreOfParentheses(char* s) {
2     int score = 0;
3     int depth = 0;
4
5
6     for (int i=0; s[i] != '\0'; i++){
7         if(s[i] == '(') {
8             depth++;
9         } else {
10            depth--;
11            if (s[i - 1] == '('){
12                score += 1 << depth;
13            }
14        }
15    }
16
17    return score;
18 }
```

Saved to local

Ln 11, Col 30

Testcase | >_ Test Result

Accepted

Runtime: 0 ms

• Case 1

• Case 2

• Case 3

Input

s =

"()"

Output

856. Score of Parentheses

Medium Topics Companies

Given a balanced parentheses string s , return *the score of the string*.

The **score** of a balanced parentheses string is based on the following rule:

- "()" has score 1.
- AB has score A + B, where A and B are balanced parentheses strings.
- (A) has score 2 * A, where A is a balanced parentheses string.

Example 1:

Input: $s = "()"$
Output: 1

Example 2:

Input: $s = "(())"$
Output: 2

Example 3:

Input: $s = "()()"$
Output: 2

</> Code

C Auto

```
1 int scoreOfParentheses(char* s) {
2     int score = 0;
3     int depth = 0;
4
5
6     for (int i=0; s[i] != '\0'; i++){
7         if(s[i] == '(') {
8             depth++;
9         } else {
10            depth--;
11            if (s[i - 1] == '('){
12                score += 1 << depth;
13            }
14        }
15    }
16
17    return score;
18 }
```

Saved to local

Ln 11, Col 30

☑️ Testcase >_ Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

$s =$
"()()"

Output



Accepted

user6837EV submitted at Feb 13, 2024 23:30

Editorial

Solution

Runtime

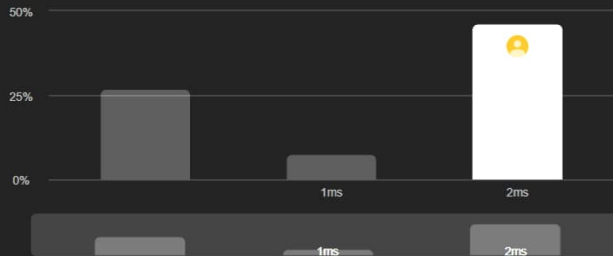
3 ms

Beats 19.23% of users with C

Memory

5.61 MB

Beats 46.15% of users with C



Code | C

```
int scoreOfParentheses(char* s) {
    int score = 0;
    int depth = 0;

    for (int i=0; s[i] != '\0'; i++){
        if(s[i] == '(') {
            depth++;
        }
    }
}
```

</> C

Submit Ctrl Enter

C v Auto

≡ □ {} ↺

```
1 int scoreOfParentheses(char* s) {
2     int score = 0;
3     int depth = 0;
4
5
6     for (int i=0; s[i] != '\0'; i++){
7         if(s[i] == '(') {
8             depth++;
9         } else {
10            depth--;
11            if (s[i-1] == '('){
12                score += 1 << depth;
13            }
14        }
15    }
16
17    return score;
18 }
```

Saved to local

Ln 11, Col 30

Testcase | Test Result

Accepted

Runtime: 0 ms

• Case 1

• Case 2

• Case 3

Input

s =

"()()"

Output