

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <numeric>
5 using namespace std;
6
7 vector<vector<int>> getLocations(string str, string subStr);
8 vector<vector<int>> collapse(vector<vector<int>> locations);
9 string underscoreify(string str, vector<vector<int>> locations);
10
11 // O(n*m) | O(n) space
12 string underscoreifySubstring(string str, string subStr) {
13     vector<vector<int>> locations = collapse(getLocations(str, subStr));
14     return underscoreify(str, locations);
15 }
16
17 vector<vector<int>> getLocations(string str, string subStr) {
18     vector<vector<int>> locations{};
19     int startIdx = 0;
20     while (startIdx < str.length()) {
21         int nextIdx = str.find(subStr, startIdx);
22         if (nextIdx != string::npos) {
23             locations.push_back(vector<int>{nextIdx, int(nextIdx + subStr.length())});
24             startIdx = nextIdx + 1;
25         } else {
26             break;
27         }
28     }
29     return locations;
30 }
31
32 vector<vector<int>> collapse(vector<vector<int>> locations) {
33     if (locations.empty()) {
```

Solution 1

Solution 2

Solution 3

```
1 using namespace std;
2
3 string underscoreifySubstring(string str, string subStr) {
4     // Write your code here.
5     return "";
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <numeric>
5 using namespace std;
6
7 vector<vector<int>> getLocations(string str, string subStr);
8 vector<vector<int>> collapse(vector<vector<int>> locations);
9 string underscoreify(string str, vector<vector<int>> locations);
10
11 // O(n*m) | O(n) space
12 string underscoreifySubstring(string str, string subStr) {
13     vector<vector<int>> locations = collapse(getLocations(str, subStr));
14     return underscoreify(str, locations);
15 }
16
17 vector<vector<int>> getLocations(string str, string subStr) {
18     vector<vector<int>> locations{};
19     int startIdx = 0;
20     while (startIdx < str.length()) {
21         int nextIdx = str.find(subStr, startIdx);
22         if (nextIdx != string::npos) {
23             locations.push_back(vector<int>{nextIdx, int(nextIdx + subStr.length())});
24             startIdx = nextIdx + 1;
25         } else {
26             break;
27         }
28     }
29     return locations;
30 }
31
32 vector<vector<int>> collapse(vector<vector<int>> locations) {
33     if (locations.empty()) {
```

```
1 using namespace std;
2
3 string underscoreifySubstring(string str, string subStr) {
4     // Write your code here.
5     return "";
6 }
7
```

