

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <numeric>
5 using namespace std;
6
7 char getNewLetter(char letter, int key, string alphabet);
8
9 // O(n) time | O(n) space
10 string caesarCypherEncryptor(string str, int key) {
11     vector<char> newLetters;
12     int newKey = key % 26;
13     string alphabet = "abcdefghijklmnopqrstuvwxyz";
14     for (int i = 0; i < str.length(); i++) {
15         newLetters.push_back(getNewLetter(str[i], newKey, alphabet));
16     }
17     return string(newLetters.begin(), newLetters.end());
18 }
19
20 char getNewLetter(char letter, int key, string alphabet) {
21     int newLetterCode = alphabet.find(letter) + key;
22     return newLetterCode <= 25 ? alphabet[newLetterCode]
23         : alphabet[-1 + newLetterCode % 25];
24 }
25
```

Solution 1Solution 2Solution 3

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

Our Tests

```
1 // Test 1: Basic encryption
2 // Input: str = "abc", key = 3
3 // Output: "def"
4
5 // Test 2: Wrapping around
6 // Input: str = "zabc", key = 3
7 // Output: "cdef"
8
9 // Test 3: Negative key
10 // Input: str = "abc", key = -3
11 // Output: "xyz"
```

Custom Output

Submit Code

```
1 // Custom Output
2
3 // Submit Code
```

