Our Solution(s) Run Code

Solution 1 Solution 2

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// O(n) time | O(n) space
func CaesarCipherEncryptor(str string, key int) string {
  shift, offset := rune(key%26), rune(26)
  runes := []rune(str)
  for i, char := range runes {
    if char >= 'a' && char+shift <= 'z' {
      char += shift
    } else {
      char += shift - offset
    }
    runes[i] = char
  }
  return string(runes)
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func CaesarCipherEncryptor(str string, key int) string {
  // Write your code here.
  return ""
}

```

Our Tests

Custom Output Submit Code

```
}

func (s *TestSuite) TestCase23(t *testing.T) {
    expected := "abc"
    output := CaesarCipherEncryption("abc", 0)
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase24(t *testing.T) {
    expected := "bcd"
    output := CaesarCipherEncryption("abc", 1)
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase25(t *testing.T) {
    expected := "xyz"
```

Run or submit code when you're ready.