

Problem Statement

Bhargava S Saveetha Engineering College

Rotate a String Left by K Positions

Write a program to rotate a given string s left by k positions without using any built-in string functions.

For example, rotating "abcdef" by 2 would give "cdefab".

Instructions: Do not use built-in rotation or substring functions. Implement the logic manually.

Solution:

```
package main

import (
    "fmt"
)

func rotateKPositionsLeft(s string, k int) string {
    stringArray := []rune(s)
    n := len(stringArray)

    k = k % n

    rotated := make([]rune, n)
    for i := 0; i < n; i++ {
        rotated[i] = stringArray[(i+k)%n]
    }

    return string(rotated)
}
```

```

func main() {

    s1 := "abcdef"

    k1 := 2

    result1 := rotateKPositionsLeft(s1, k1)

    fmt.Println("Test case 1 - Rotated string:", result1)

    s2 := "abcdef"

    k2 := 9

    result2 := rotateKPositionsLeft(s2, k2)

    fmt.Println("Test case 2 - Rotated string:", result2)

    s3 := "abcdef"

    k3 := 6

    result3 := rotateKPositionsLeft(s3, k3)

    fmt.Println("Test case 3 - Rotated string:", result3)

}

```

Output:

Test case 1 - Rotated string: cdefab

Test case 2 - Rotated string: defabc

Test case 3 - Rotated string: abcdef

```

main.go
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func rotateKPositionsLeft(s string, k int) string {
8     stringArray := []rune(s)
9     n := len(stringArray)
10
11     k = k % n
12
13     rotated := make([]rune, n)
14     for i := 0; i < n; i++ {
15         rotated[i] = stringArray[(i+k)%n]
16     }
17
18     return string(rotated)
19 }
20
21 func main() {
22
23     s1 := "abcdef"
24     k1 := 2

```

```

go run /tmp/1w1x0sY2Fc.go
Test case 1 - Rotated string: cdefab
Test case 2 - Rotated string: defabc
Test case 3 - Rotated string: abcdef

```