problem statement
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.

program code

```
package main

import (
	"fmt"
)

func rotateStringLeft(originalString string, pos int) string {
	strLen := len(originalString)

	if strLen == 0 {
		return originalString
	}
	pos = pos % strLen

	rotStr := make([]byte, strLen)

	for idx := 0; idx < strLen; idx++ {
		newIdx := (idx + strLen - pos) % strLen
		rotStr[newIdx] = originalString[idx]
	}

	finalRes := ""
	for _, ch := range rotStr {
		finalRes += string(ch)
	}

	return finalRes
}

func main() {
	var inStr string
	var rotatePos int
	fmt.Print("String:")
	fmt.Scanln(&inStr)
	fmt.Print("Position:")
	fmt.Scanln(&rotatePos)
	rotRes := rotateStringLeft(inStr, rotatePos)
	fmt.Println("Rotated String:", rotRes)
}
```

Output:

```
    }
    pos = pos % strLen

    rotStr := make([]byte, strLen)

    for idx := 0; idx < strLen; idx++ {
        newIdx := (idx + strLen - pos) % strLen
        rotStr[newIdx] = originalString[idx]
    }

    finalRes := ""
    for _, ch := range rotStr {
        finalRes += string(ch)
    }

    return finalRes
}

func main() {
    var inStr string
    var rotatePos int
    fmt.Print("String:")
    fmt.Scanln(&inStr)
    fmt.Print("Position:")
    fmt.Scanln(&rotatePos)
    rotRes := rotateStringLeft(inStr, rotatePos)
    fmt.Println("Rotated String:", rotRes)
}

```

input

```
String:abcdef
Position:2
Rotated String: cdefab
```

```
String:hashagile
Position:5
Rotated String: gilehasha
```

```
String:knowledge
Position:1
Rotated String: nowledgek
```