Autor: Eduardo Murillo

## Programación de Sistemas

### Taller 3 - Git y C

#### Introducción

En este taller practicarán el uso de Git, y a la vez practicaran la sintaxis básica de C para resolver problemas.

### Parte 1

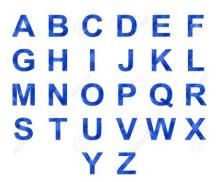
Para esta práctica implementaran dos programas, detallados a continuación:

Uno de los algoritmos de encriptación más básicos son los cifrados de sustitución el **cifrado cíclico.** En este esquema, cada letra de la palabra a encriptar se sustituye por una letra **n** veces delante de esta; a **n** le llamamos la llave numérica. Por ejemplo, si usamos llave numérica 3 y el mensaje a cifrar es "hola pedro"

Mensaje: hola pedro Cifrado llave 3: krod shgur

Escriba un programa, que reciba un mensaje, y la llave numérica, y devuelva el mensaje cifrado.

Si la llave numérica es mayor al número de letras delante del carácter que queremos codificar, entonces damos la vuelta y empezamos desde el inicio del alfabeto hasta completar el cifrado Por ejemplo:



Si el mensaje es 'ZOILA', y la llave numérica es 3, en mensaje codificado seria:

**CRLOD** 

La llave numérica puede ser negativa también. En este caso reemplazamos la letra por una letra –n veces atrás.

Autor: Eduardo Murillo

#### Consideraciones:

- 1. Se respetan las mayúsculas y minúsculas
- 2. Los signos de puntuación, saltos de líneas y espacios quedan iguales.
- 3. Use el alfabeto en INGLES (26 letras).
- 4. Implemente el algoritmo de cifrado como una función.
- 5. USE git para llevar control de los cambios en su código. Debe tener al menos TRES commits. Este será el branch master.
- 6. No olvide considerar y manejar posibles condiciones de error.

La salida de su programa será:

```
Cifrado cíclico

Ingrese mensaje a cifrar: hola pedro

Ingrese la llave numérica: 3

Mensaje cifrado: krod shgur
```

#### Parte 2

En esta parte, Ud. creará un nuevo **branch** e implementará el siguiente los siguientes cambios:

1. Cambiará el programa para que pueda correr de manera no interactiva o interactiva. Si programa NO recibe argumentos, funcionará como en la parte uno. Si recibe dos argumentos, inmediatamente dará el resultado. Los argumentos son la llave numérica y el string a codificar (al dos commits para estar parte)

## \$ ./programa 3 "hola pedro"

```
Mensaje cifrado: krod shgur
```

2. Implemente una función que permita ver el mensaje cifrado en clave morse. Los espacios son reemplazados por un slash. Un ejemplo y la tabla de código morse se muestra a continuación (al menos dos commits en esta parte):

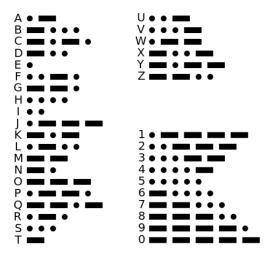
# \$ ./programa 3 "hola pedro"

```
Mensaje cifrado: krod shgur

Mensaje cifrado en morse: -.- .-. / ... / ... --. .-.
```

### International Morse Code

- The length of a dot is one unit.
   A dash is three units.
   The space between parts of the same letter is one unit.
   The space between letters is three units.
   The space between words is seven units.



### Parte 3

En este parte, Ud. reconciliará las dos ramas master (parte1) y la rama de la parte 2 (merge). Resuelva los conflictos de ser necesario. El resultado debe ser que tanto en modo interactivo como en modo no interactivo, se muestre el mensaje cifrado y en morse:

## \$./prog

```
Cifrado cíclico
Ingrese mensaje a cifrar: hola pedro
Ingrese la llave numérica: 3
Mensaje cifrado: krod shgur
Mensaje cifrado en morse: -.- .-. / ... / ... --. .-.
$./prog 3 "hola pedro"
Mensaje cifrado: krod shgur
```

Mensaje cifrado en morse: -.- .-. / ... / ... --. .-.

## **Entregable**

URL del repositorio de git. Fecha de entrega 26 de Mayo de 2017, 23:59. Tarea Individual