Taller 2 - Cifrado Cesar

P1: Implementación Paso a Paso (Cifrado simple)

En esta parte se va a usar el alfabeto [A-Z] todo en mayúsculas el código esta nombrado en la impresión y en el código con el mismo para a la hora de poder ejecutarse se pueda revisar cual es el que corresponde el código, se debe comentar cada código cuando se termine de ejecutar para que no haya interferencias con el siguiente cabe resaltar que cada código está separado son 3 códigos para el primer punto inician desde el 1.1 – 1.4 en el último punto se desprenden 3 de los cuales quedaron por separados dándonos un total de 6 códigos implementados para la primera parte.

1.1 Cifrar una Letra

- Construir una función que tome una letra y una clave, y retorne la letra cifrada, en este código se creó una función para que recorra cada uno de los elementos relacionados en el arreglo y se cifre bajo una clave escogida para este ejemplo se usó la “A,23” de estilo alfanumérico y al final nos retornara una letra que se encuentre ubicada en la posición especifica.

```
#cifrado de letra#

array="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
clave=("A,23")

def alfabeto(clave,C):

    idx = array.index(C)
    letracif=array[(idx+clave%len(array))]
    return letracif
```

```
#-----------------------------------------------------#

#cifrado de letra#
#alfabeto(3,"A")
#print(alfabeto(3,"A"))
```

1.2 Cifrar un Mensaje Completo

- Crear una función que use la función previamente creada para cifrar un mensaje completo, en este código se sigue el mismo parámetro de la creación del arreglo y la lógica para el uso de la función del alfabeto solo que en este punto se creará un mensaje que será cifrado bajo una clave la cual es numérica y es 23, se estipulo un mensaje ya escogido para que a la hora de cifrarlo se recorra cada una de las letras que están en él por medio del uso de la función cifrar mensaje que tiene un ciclo for para realizar el recorrido de cada carácter y se verifica si cada letra corresponde a las que se ingresaron en el arreglo y pueda devolvernos el mensaje de manera cifrada.

```python
#Cifrado de mensaje#

array = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"

def alfabeto(clave, C):
    idx = array.index(C)
    letracif = array[(idx + clave) % len(array)]
    return letracif

def cifrar_mensaje(clave, mensaje):
    mensaje_cifrado = ""
    for letra in mensaje:
        if letra in array:
            mensaje_cifrado += alfabeto(clave, letra)
        else:
            mensaje_cifrado += letra  # Si la letra no está en el alfabeto, se añade tal cual
    return mensaje_cifrado

mensaje = "ENELCIELOAZULDEUNDIACLAROBAILANLOSSUEÑOSLIBRESYRAROSCONELVIENTOSUSURRANSECRETOSENCADARINCONNUEVOSRETOS"
clave = 23
```

```python
#Cifrado del mensaje
#mensaje_cifrado = cifrar_mensaje(clave, mensaje)
#print(mensaje_cifrado)
```

1.3 Descifrar un Mensaje

- Implementar una función que permita descifrar el mensaje indicándole la clave, en este código nos permite descodificar el mensaje que teníamos anteriormente por medio del ingreso de la clave otorgada realizando el ciclo de verificación de los caracteres por medio de la función descifrar que toca en cuenta el arreglo ya creado y la verificación de los caracteres del mensaje y retorna el mensaje.

```python
#descifrado de mensaje#

array = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
clave = 23
mensaje = "ENELCIELOAZULDEUNDIACLAROBAILANLOSSUEÑOSLIBRESYRAROSCONELVIENTOSUSURRANSECRETOSENCADARINCONNUEVOSRETOS"

def descifrar(clave, mensaje):
    array = "mensaje"
    s = ""
    for i in mensaje:
        if i in array:
            idx = array.index(i)
            frasecif = array[(idx - clave) % len(array)]
            s += frasecif
        else:
            s += i
    return s
```

```python
#Descifrado del mensaje
mensaje_descifrado = descifrar(clave, mensaje)
print(mensaje_descifrado)
```

1.4 Pruebas Básicas

- **Cifra "HOLA" con clave 3 → "KROD"**, en este código se realizó la revisión de la prueba del código de cifrar la palabra HOLA con la clave 3 y retornándonos la palabra KROD donde se creó una función donde se recorriera cada una de las letras y las posiciones de la palabra dentro del arreglo creado y que devolviese la palabra KROD al imprimir de manera cifrada.

```python
#cifrado de mensaje hola#

array = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
clave = 3
palabra = "HOLA"

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave) % len(array)]

def cifrar_palabra(clave, mensaje):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra in array:
            palabra_cifrada += alfabeto(clave, letra)
        else:
            palabra_cifrada += letra
    return palabra_cifrada      You, hace 31 minutos •
```

```python
#Cifrado de mensaje hola
palabra_cifrada = cifrar_palabra(clave, palabra)
print(palabra_cifrada)
```

- **Descifra "CDE" con clave 5 → "ZYX"** en este código se implementó la función descifrar_mensaje para que las letras cifradas nos devolvieran una información recorriendo el arreglo y el orden de cada una de ellas con una clave 5 retomando la función del alfabeto para recorrer cada una de las letras que se encuentran guardadas dentro del arreglo.

```python
#descifrado de mensaje cde#

array = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
clave = 5
mensaje = "CDE"

def alfabeto_descifrar(clave, letra):
    posicion = array.index(letra)
    return array[(posicion - clave) % len(array)]

def descifrar_mensaje(clave, mensaje):
    mensaje_descifrado = ""
    for letra in mensaje:
        if letra in array:
            mensaje_descifrado += alfabeto_descifrar(clave, letra)
        else:
            mensaje_descifrado += letra
    return mensaje_descifrado
```

```python
#descifrado de mensaje cde
mensaje_descifrado = descifrar_mensaje(clave, mensaje)
print(mensaje_descifrado)       You, hace 1 segundo • U
```

- **Probar con espacios y símbolos: "¡Hola Mundo!" → "¡KROD PXQGR!"**, aquí se aplica las mismas funciones para cifrar el mensaje como en el primer punto de prueba que era creación de la palabra cifrada de hola solo que en este punto se le agrega un complemento al mensaje de otra palabra la cual es MUNDO y los signos de interrogación y el espacio entre las 2 palabras así mismo se recorrerá el ciclo el ciclo y se realizara la revisión de los caracteres que se están ingresando en el mensaje y que están determinados bajo unos parámetros que están en el arreglo si cumple la descripción nos devolverá la palabra ¡KROD PXQGR! En la impresión de manera que sea en mayúscula cada carácter de la palabra gracias a la implementación del upper y por medio del isupper se revisa si la cadena de texto está en mayúsculas para así mismo devolver los parámetros ingresados.

```python
#Cifrado de mensaje ¡Hola Mundo!#

array = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
clave = 3
mensaje = "¡HOLA MUNDO!"

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave) % len(array)]

def cifrar_palabra(clave, mensaje):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra.upper() in array:
            if letra.isupper():
                palabra_cifrada += alfabeto(clave, letra.upper())
            else:
                palabra_cifrada += alfabeto(clave, letra.upper()).lower()
        else:
            palabra_cifrada += letra
    return palabra_cifrada      You, hace 43 minutos • ACTUALIZAR
```

```python
#Cifrado de mensaje ¡Hola Mundo!
cifrar_palabra = cifrar_palabra(clave, mensaje)
print(cifrar_palabra)
```

P2: Implementación Paso a Paso (Cifrado extendido)

2.1 Integrando mayúsculas y minúsculas [Con pruebas]

- Al alfabeto que se tiene ABC...XYZ se le van a agregar las minúsculas de la forma ABC...XYZabc...xyz, **Cifra "HOLA" con clave 3 → "KROD",** en este código se realizó la revisión de la prueba del código de cifrar la palabra HOLA con la clave 3 y retornándonos la palabra KROD donde se creó una función donde se recorriera cada una de las letras y las posiciones de la palabra dentro del arreglo creado y que devolviese la palabra KROD al imprimir de manera cifrada agregándole el abecedario en minúsculas.

```
# Cifrado de palabra

array = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
clave = 3
mensaje = "HOLA"

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave) % len(array)]

def cifrar_palabra(mensaje, clave):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra in array:
            palabra_cifrada += alfabeto(clave, letra)
        else:
            palabra_cifrada += letra
    return palabra_cifrada
```

```
# Cifrado de palabra
#mensaje_cifrado = cifrar_palabra(mensaje, clave)
#print(mensaje_cifrado)
```

- **Descifra "CDE" con clave 5 → "ZYX"** en este código se implementó la función descifrar_mensaje para que las letras cifradas nos devolvieran una información recorriendo el arreglo y el orden de cada una de ellas con una clave 5 retomando la función del alfabeto para recorrer cada una de las letras que se encuentran guardadas dentro del arreglo agregando el abecedario en minúsculas.

```python
#Descifrado de mensaje#

array = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
clave = 5
mensaje = "CDE"

def alfabeto_descifrar(clave, letra):
    posicion = array.index(letra)
    return array[(posicion - clave) % len(array)]

def descifrar_mensaje(clave, mensaje):
    mensaje_descifrado = ""
    for letra in mensaje:
        if letra in array:
            mensaje_descifrado += alfabeto_descifrar(clave, letra)
        else:
            mensaje_descifrado += letra
    return mensaje_descifrado       You, hace 1 segundo • Uncommitte
```

```python
# Descifrado del mensaje
mensaje_descifrado = descifrar_mensaje(clave, mensaje)
print(mensaje_descifrado)
```

- **Probar con espacios y símbolos: "¡Hola Mundo!" → "¡KROD PXQGR!"**, aquí se aplica las mismas funciones para cifrar el mensaje como en el primer punto de prueba que era creación de la palabra cifrada de hola solo que en este punto se le agrega un complemento al mensaje de otra palabra la cual es MUNDO y los signos de interrogación y el espacio entre las 2 palabras así mismo se recorrerá el ciclo el ciclo y se realizara la revisión de los caracteres que se están ingresando en el mensaje y que están determinados bajo unos parámetros que están en el arreglo si cumple la descripción nos devolverá la palabra ¡KROD PXQGR! agregando el abecedario en minúsculas.

```python
#cifrado de mensaje ¡HOLA MUNDO!#

array = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
mensaje = "¡HOLA MUNDO!"
clave = 3

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave) % len(array)]

def cifrar_palabra(mensaje):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra in array:
            palabra_cifrada += alfabeto(clave, letra)
        else:
            palabra_cifrada += letra
    return palabra_cifrada      You, hace 1 hora • ACTUALIZAR
```

```python
# Cifrado de mensaje ¡HOLA MUNDO!
mensaje_cifrado = cifrar_palabra(mensaje)
print(mensaje_cifrado)
```

2.2 Intercalando mayúsculas y minúsculas [Con prubas]

- Ahora el alfabeto debe de seguir la forma AaBbCc...XxYyZz, **Cifra "HOLA" con clave 3 → "KROD"**, en este código se realizó la revisión de la prueba del código de cifrar la palabra HOLA con la clave 3 y retornándonos la palabra KROD donde se creó una función donde se recorriera cada una de las letras y las posiciones de la palabra dentro del arreglo creado y que devolviese la palabra KROD al imprimir de manera cifrada de manera que sea en mayúscula cada carácter de la palabra gracias a la implementación del upper y haciendo la revisión del índice se revisa si la cadena de texto está en mayúsculas para así mismo devolver los parámetros ingresados, adicional a esto se le coloca el 2 para que se recorra de doble forma teniendo en cuenta que las letras se encuentran de manera repetida solo que el formato cambia dado que se encuentran intercaladas.

```python
# Cifrado de palabra intercalada

array = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"
clave = 3
palabra = "HOLA"

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave * 2) % len(array)]

def cifrar_palabra(clave, mensaje):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra.upper() in array.upper():
            letra_array = array[array.upper().index(letra.upper())]
            palabra_cifrada += alfabeto(clave, letra_array)
        else:
            palabra_cifrada += letra
    return palabra_cifrada.upper()
```

```python
# Cifrado de palabra intercalada
cifrado = cifrar_palabra(clave, palabra)
print(cifrado)       You, hace 1 segundo •
```

- **Descifra "CDE" con clave 5 → "ZYX"** en este código se implementó la función descifrar_mensaje para que las letras cifradas nos devolvieran una información recorriendo el arreglo y el orden de cada una de ellas con una clave 5 retomando la función del alfabeto para recorrer cada una de las letras que se encuentran guardadas dentro del arreglo agregando el abecedario en minúsculas, adicional a esto se le coloca el 2 para que se recorra de doble forma teniendo en cuenta que las letras se encuentran de manera repetida solo que el formato cambia dado que se encuentran intercaladas.

```python
# Descifrado del mensaje intercalado

array = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"
clave = 5
mensaje = "CDE"

def alfabeto_descifrar(clave, letra):
    posicion = array.index(letra)
    return array[(posicion - clave * 2) % len(array)]

def descifrar_mensaje(clave, mensaje):
    mensaje_descifrado = ""
    for letra in mensaje:
        if letra in array:
            mensaje_descifrado += alfabeto_descifrar(clave, letra)
        else:
            mensaje_descifrado += letra
    return mensaje_descifrado      You, hace 2 horas • ACTUALIZAR
```

```python
# Descifrado del mensaje intercalado
#mensaje_descifrado = descifrar_mensaje(clave, mensaje)
#print(mensaje_descifrado)
```

- **Probar con espacios y símbolos: "¡Hola Mundo!" → "¡KROD PXQGR!"**, aquí se aplica las mismas funciones para cifrar el mensaje como en el primer punto de prueba que era creación de la palabra cifrada de hola solo que en este punto se le agrega un complemento al mensaje de otra palabra la cual es MUNDO y los signos de interrogación y el espacio entre las 2 palabras así mismo se recorrerá el ciclo el ciclo y se realizara la revisión de los caracteres que se están ingresando en el mensaje y que están determinados bajo unos parámetros que están en el arreglo si cumple la descripción nos devolverá la palabra ¡KROD PXQGR!, adicional a esto se le coloca el 2 para que se recorra de doble forma teniendo en cuenta que las letras se encuentran de manera repetida solo que el formato cambia dado que se encuentran intercaladas.

```python
# Cifrado de mensaje ¡HOLA MUNDO! intercalado

array = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"
clave = 3
mensaje = "¡HOLA MUNDO!"

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave * 2) % len(array)]

def cifrar_palabra(mensaje):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra in array:
            palabra_cifrada += alfabeto(clave, letra)
        else:
            palabra_cifrada += letra
    return palabra_cifrada      You, hace 2 horas • ACTUALIZAR
```

```python
# Cifrado de mensaje ¡HOLA MUNDO! intercalado
#cifrar_palabra = cifrar_palabra (mensaje)
#print(cifrar_palabra)
```

2.3 Usando simbolos [Con prubas]

- Ahora al alfabeto se le debe de agregar los símbolos, de la forma AaBbCc...XxYyZz!"#$%&'()*+,-./:;<=>?@[\\]^_{|}~`, **Cifra "HOLA" con clave 3 → "KROD"**, en este código se realizó la revisión de la prueba del código de cifrar la palabra HOLA con la clave 3 y retornándonos la palabra KROD donde se creó una función donde se recorriera cada una de las letras y las posiciones de la palabra dentro del arreglo creado y que devolviese la palabra KROD al imprimir de manera cifrada, adicional a esto se le coloca el 2 para que se recorra de doble forma teniendo en cuenta que las letras se encuentran de manera repetida solo que el formato cambia dado que se encuentran intercaladas, se le agregaron los símbolos al código.

```python
# Cifrar palabra y simbolos

array = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz!@#$%^&*()_+-=[]{}|;:'\",.<>?/~`"
clave = 3
mensaje = "HOLA"

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave*2) % len(array)]

def cifrar_palabra(mensaje, clave):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra in array:
            palabra_cifrada += alfabeto(clave, letra)
        else:
            palabra_cifrada += letra
    return palabra_cifrada
```

```python
# Cifrar palabra y simbolos
#cifrado = cifrar_palabra(mensaje, clave)
#print(cifrado)      You, hace 1 segundo
```

- **Descifra "CDE" con clave 5 → "ZYX"** en este código se implementó la función descifrar_mensaje para que las letras cifradas nos devolvieran una información recorriendo el arreglo y el orden de cada una de ellas con una clave 5 retomando la función del alfabeto para recorrer cada una de las letras que se encuentran guardadas dentro del arreglo agregando el abecedario en minúsculas, adicional a esto se le coloca el 2 para que se recorra de doble forma teniendo en cuenta que las letras se encuentran de manera repetida solo que el formato cambia dado que se encuentran intercaladas, se le agregaron los símbolos al código.

```python
# Descifrar el mensaje y simbolos

array = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz!@#$%^&*()_+-=[]{}|;:'\",.<>?/~`"
clave = 5
mensaje = "CDE"

def alfabeto_descifrar(clave, letra):
    posicion = array.index(letra)
    return array[(posicion - clave) % len(array)]

def descifrar_mensaje(clave, mensaje):
    mensaje_descifrado = ""
    for letra in mensaje:
        if letra in array:
            mensaje_descifrado += alfabeto_descifrar(clave, letra)
        else:
            mensaje_descifrado += letra
    return mensaje_descifrado
```

```python
# Descifrar el mensaje y simbolos
#mensaje_descifrado = descifrar_mensaje(clave, mensaje)
#print(mensaje_descifrado)
```

- **Probar con espacios y símbolos: "¡Hola Mundo!" → "¡KROD PXQGR!"**, aquí se aplica las mismas funciones para cifrar el mensaje como en el primer punto de prueba que era creación de la palabra cifrada de hola solo que en este punto se le agrega un complemento al mensaje de otra palabra la cual es MUNDO y los signos de interrogación y el espacio entre las 2 palabras así mismo se recorrerá el ciclo el ciclo y se realizara la revisión de los caracteres que se están ingresando en el mensaje y que están determinados bajo unos parámetros que están en el arreglo si cumple la descripción nos devolverá la palabra ¡KROD PXQGR!, adicional a esto se le coloca el 2 para que se recorra de doble forma teniendo en cuenta que las letras se encuentran de manera repetida solo que el formato cambia dado que se encuentran intercaladas, se le agregaron los símbolos al código.

```python
# Cifrar el mensaje y simbolos

array = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz!@#$%^&*()_+-=[]{}|;:'\",.<>?/~`"
mensaje = "¡HOLA MUNDO!"
clave = 3

def alfabeto(clave, letra):
    posicion = array.index(letra)
    return array[(posicion + clave*2) % len(array)]

def cifrar_palabra(mensaje):
    palabra_cifrada = ""
    for letra in mensaje:
        if letra in array:
            palabra_cifrada += alfabeto(clave, letra)
        else:
            palabra_cifrada += letra
    return palabra_cifrada
```

```python
# Cifrar el mensaje y simbolos
#cifrado = cifrar_palabra(mensaje)
#print(cifrado)      You, hace 1 s
```

Preguntas para Reflexión:

- ¿Por qué el cifrado César es inseguro hoy en día?

Dado que se basa en puntos fijos donde se guarda la información se torna muy fácil en descifrar por la corta seguridad que tiene el cifrado cesar, las letras se mueven recorriendo el mismo orden siempre en el mismo sentido que los símbolos, números y caracteres que se implementen en el mensaje a cifrar.

- ¿Cómo mejorarías este algoritmo para hacerlo más robusto?

Implementaría que el sistema intercambiara posiciones con cualquier carácter que se ingrese en los arreglos de código para cifrar la información, implementaría el cambio de letras con números y símbolos en cuanto a las posiciones a recorrer cuando se este cifrando es decir que la letra puede ser número, símbolo, letras al mismo tiempo y que se pudieran guardar en el mismo lugar, incrementación en las claves para que sean mas extensas.