

## Documentación

```
import string

#ejercicio 1.1 Cifrar una Letra
abecedarioMay= string.ascii_uppercase # trae todo el abecedario

def cifrarletra(letra, clave):
    indiceLetra= abecedarioMay.index(letra) #busca el indice de la letra y
    lo almacena en la variable indiceLetra
    return abecedarioMay[(indiceLetra+clave)%26] #se hace el cambio de la
    letra sumando la clave que moverá el indice y con el % que es el modulo para
    este caso seria 26 que son las mayusculas

#se realiza la impresión por pantalla
print("*****")
print("Ejercicio 1.1 Cifrar una Letra")
print(cifrarletra("A", 3))
print("*****\n")
```

### RESULTADO

```
*****
Ejercicio 1.1 Cifrar una Letra
D
*****
```

### #ejercicio 1.2 Cifrar un Mensaje Completo

```
def mensajeCompleto(frase, clave):    # Convertir todo a mayúsculas para
    evitar errores
    frase = frase.upper() #si el usuario digita en minúscula la cambiara a
    mayuscula
    frase_cifrada = "" #la variable que se comportara como array para
    almacenar la nueva contraseña

    for letra in frase:# se usara el for para que se desplace letra por
    letra en la frase que el usuario digitará
        if letra in abecedarioMay: #se usa la condición IF si la encuentra
        dentro de la base del abecedario pasara a cambiarla en la siguiente línea
            frase_cifrada += cifrarletra(letra, clave) # almacena la frase
    completa
    else:
```

```

        frase_cifrada += letra #almacena el símbolo que no este en la
variable del abecedario
    return frase_cifrada # Devuelve la frase cifrada completa

```

```

#PRUEBAS

```

```

print("*****")
print("Ejercicio 1.2 Cifrar un Mensaje Completo")
print(mensajeCompleto("CONTRASEÑA SEGURA",3))
print("*****\n")

```

```

*****

```

```

Ejercicio 1.2 Cifrar un Mensaje Completo
FRQWUDVHÑD VHJXUD
*****

```

```

#Ejercicio 1.3 descifrar codigo

```

```

def descifrar_frase(frase, clave):
    abecedarioMayMin = string.ascii_letters #aquí se usara Mayúsculas y
minúsculas
    frase_descifrada = "" # Variable para almacenar la frase cifrada

    for letra in frase: # se usa el for para recorrer letra por letra de la
frase que el usuario digitó
        if letra in abecedarioMay:#si la letra la encuentra en el abecedari
cambiala con el siguiente código
            indiceLetra = (abecedarioMay.index(letra) - clave) % 26
#transforma la letra según la clave digitada
            frase_descifrada += abecedarioMay[indiceLetra] #almacena la
letra cambiada
        else:
            frase_descifrada += letra # si no esta en el abecedario es un
carácter y la almacena sin cambios

    print(frase_descifrada)#imprime la función

print("Ejercicio 1.3 Descifrar un Mensaje")
descifrar_frase("FRQWUDVHÑD VHJXUD", 3)
print("*****\n")

```

```

Ejercicio 1.3 Descifrar un Mensaje
CONTRASEÑA SEGURA
*****

```

```
#Ejercicio 1.4 pruebas
print("Ejercicio 1.4 Pruebas")
print(mensajeCompleto("HOLA",3))
descifrar_frase("CDE", 5)
print(mensajeCompleto("¡Hola Mundo!",3))
```

```
Ejercicio 1.4 Pruebas
KROD
XYZ
¡KROD PXQGR!
```

```
#P2: Implementación Paso a Paso (Cifrado extendido)
```

```
#3.1 Integrando mayúsculas y minúsculas [Con pruebas]
```

```
def cifrar_frase(frase, clave):
    abecedarioMayMin =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" #se crea una nueva
base a string con Mayúsculas y minúsculas
    frase_cifrada = "" # Variable para almacenar la frase cifrada

    for letra in frase:
        if letra in abecedarioMayMin:#si esta dentro del abecedario procede
a cambiarla con la siguiente línea
            indiceLetra = (abecedarioMayMin.index(letra) + clave) % 52
#transforma la letra a la nueva letra con un índice de que 52 es la cantidad
de letras con mayúsculas y minúsculas en total.
            frase_cifrada += abecedarioMayMin[indiceLetra] #almacena la
letra cambiada
        else:
            frase_cifrada += letra # si no esta en el abecedario es un
carácter

    print(frase_cifrada)

print("\n*****ALFABETO COMPLETO*****")
print("3.1 Integrando mayúsculas y minúsculas")
cifrar_frase("ABCDabcd", 3)
print("*****\n")
```

```
*****ALFABETO COMPLETO*****
```

```
3.1 Integrando mayúsculas y minúsculas
```

```
DEFGdefg
```

```
*****
```

```
#3.2 Intercalando mayúsculas y minúsculas
```

```
def cifrar_frase_intercalado(frase, clave):
    abecedarioMayMin =
"AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"
    frase_cifrada = "" # Variable para almacenar la frase cifrada

    for letra in frase: #recorre la frase letra por letra la frase a cambiar
        if letra in abecedarioMayMin: #si esta dentro del abecedario procede
a cambiarla
            indiceLetra = (abecedarioMayMin.index(letra) + clave) % 52
#transforma la letra a la nueva letra con 52 índices que es la cantidad de
letras con mayúsculas y minúsculas total.
            frase_cifrada += abecedarioMayMin[indiceLetra] #almacena la
letra cambiada
        else:
            frase_cifrada += letra # si no esta en el abecedario es un
carácter

    print(frase_cifrada)

print("3.2 Intercalando mayúsculas y minúsculas")
cifrar_frase_intercalado("AaBbCcDd", 3)
print("*****\n")
```

```
3.2 Intercalando mayúsculas y minúsculas
```

```
bCcDdEeF
```

```
*****
```

```

#3.3 Usando símbolos
def cifrar_frase_simbolos(frase, clave):
    abecedarioMayMinSim =
"AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz!#$%&'()*+,-
./:;<=>?@[\\]^_`{|}~`"
    frase_cifrada = "" # Variable para almacenar la frase cifrada

    for letra in frase:
        if letra in abecedarioMayMinSim: #si esta dentro del abecedario
            procede a cambiarlo en la siguiente línea
            indiceLetra = (abecedarioMayMinSim.index(letra) + clave) % 83
            #cambia la letra según la nueva clase y en este caso son 83 caracteres del
            que se compone
            frase_cifrada += abecedarioMayMinSim[indiceLetra] # almacena la
            letra cambiada
        else:
            frase_cifrada += letra # Si no está en el abecedario, lo deja
            igual

    print(frase_cifrada)

print("3.3 Usando simbolo")
cifrar_frase_simbolos("AaBbCcDd !#$%", 3)
print("*****\n")

```

```

3.3 Usando simbolo
bCcDdEeF %&'(
*****

```

Respuesta de preguntas:

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

Dentro de una fórmula matemática se podría resolver, apoyándose en la fuerza bruta

- ¿Cómo mejorarías este algoritmo para hacerlo más robusto?
- Añadiendo una operación para que al momento de tomar el índice cambie totalmente

