Desafio: Codificador de Mensagens

Você deve criar uma função chamada `codificar_mensagem` que recebe duas strings como entrada:

1. **mensagem**: a mensagem a ser codificada.
2. **chave**: uma palavra que será usada para codificar a mensagem.

O objetivo é substituir cada letra da mensagem pela letra correspondente na chave, repetindo a chave quantas vezes forem necessárias. Se a mensagem contiver caracteres que não são letras, eles devem permanecer inalterados.

Regras:

A codificação é feita com base no índice de cada caractere:

1. O primeiro caractere da mensagem será substituído pelo primeiro caractere da chave.
    - O segundo caractere da mensagem será substituído pelo segundo caractere da chave, e assim por diante.
    - Quando a chave terminar, ela será repetida.
2. Apenas substitua os caracteres alfabéticos. Outros caracteres, como números, espaços e símbolos, devem ser mantidos.
3. Não é permitido usar bibliotecas externas.

Exemplo:

In [ ]:
```
mensagem = "Segredo: Vamos ao parque às 18h!"
chave = "abc"

resultado = codificar_mensagem(mensagem, chave)

print(resultado)
```

Saída esperada:

In [ ]:
```
Sfgufdp: Wbmpt bp qbsrvf áu 18h!
```

Explicação:

- A mensagem original é `"Segredo: Vamos ao parque às 18h!"`.
- A chave repetida será `"abcabcabcabcabcabcabcabcabcab"`.
- Cada letra da mensagem foi substituída pela letra correspondente na chave, enquanto outros caracteres permaneceram inalterados.

Nível Avançado (Opcional):

Implemente também a função `decodificar_mensagem` para reverter o processo e recuperar a mensagem original.

Boa sorte! 😊

Solução

In [39]:
```
def codificar_mensagem(mensagem):
    alfabeto = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"  # Lista m
    mensagem_codificada = ""

    for caractere in mensagem:
        if caractere.isalpha():  # Verifica se é uma letra
             # Gera um índice aleatório para o alfabeto
            indice_aleatorio = (ord(caractere) + 7 + 13) % len(alfabeto)
            nova_letra = alfabeto[indice_aleatorio]
            mensagem_codificada += nova_letra
        else:
            mensagem_codificada += caractere  # Mantém os caracteres não alfabét

    return mensagem_codificada
```

In [41]:
```
# Teste
mensagem = "Segredo: Vamos ao parque às 18h!"
resultado = codificar_mensagem(mensagem)

print("Mensagem original: ", mensagem)
print("Mensagem codificada: ", resultado)
```

```
Mensagem original:  Segredo: Vamos ao parque às 18h!
Mensagem codificada:  ZrtErqB: cnzBF nB CnEDHr KF 18u!
```

Implementação de função para decodificar a mensagem codificada

In [48]:
```
def codificar_mensagem(mensagem):
    alfabeto = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
    mensagem_codificada = ""
    mapa_codificacao = {}  # Dicionário para registrar as substituições

    for caractere in mensagem:
        if caractere.isalpha():  # Verifica se é uma letra
            indice_aleatorio = (ord(caractere) * 7 + 13) % len(alfabeto)
            nova_letra = alfabeto[indice_aleatorio]

            # Adiciona ao mapa de codificação
            mapa_codificacao[caractere] = nova_letra
            mensagem_codificada += nova_letra
        else:
            mensagem_codificada += caractere  # Mantém os caracteres não alfabét
```

```
    return mensagem_codificada, mapa_codificacao


def decodificar_mensagem(mensagem_codificada, mapa_codificacao):
    mensagem_original = ""

    # inverte  dicionárop para decoficar a mensagem
    mapa_decodificacao = {value: key for key, value in mapa_codificacao.items()}

    for caractere in mensagem_codificada:
        if caractere in mapa_decodificacao:  # Substitui pela letra original
            mensagem_original += mapa_decodificacao[caractere]
        else:
            mensagem_original += caractere  # Mantém os caracteres não alfabétic

    return mensagem_original
```

In [49]:
```
# Teste
mensagem = "Segredo: Vamos ao parque às 18h!"

# Codifica a mensagem
mensagem_codificada, mapa = codificar_mensagem(mensagem)

# Mostra a mensagem codificada
print("Mensagem original:", mensagem)
print("Mensagem codificada:", mensagem_codificada)

# Decodifica a mensagem
mensagem_decodificada = decodificar_mensagem(mensagem_codificada, mapa)
print("Mensagem decodificada:", mensagem_decodificada)
```

```
Mensagem original: Segredo: Vamos ao parque às 18h!
Mensagem codificada: wSgFSLk: RqWkM qk rqFyaS vM 18n!
Mensagem decodificada: Segredo: Vamos ao parque às 18h!
```

Usando bibliotecas

In [75]:
```
import string
from random import choice


def codificar_mensagem(mensagem):
    alfabeto = string.ascii_letters  # Todas as letras do alfabeto (maiúsculas e
    mensagem_codificada = ""
    mapa_codificacao = {}  # Dicionário para registrar as substituições

    for caractere in mensagem:
        if caractere.isalpha():  # Verifica se é uma letra
            if caractere not in mapa_codificacao:
                nova_letra = choice(alfabeto)  # Escolhe uma letra aleatoriament
                # Adiciona ao mapa de codificação
                mapa_codificacao[caractere] = nova_letra
            else:
                nova_letra = mapa_codificacao[caractere]
            mensagem_codificada += nova_letra
        else:
            mensagem_codificada += caractere

    return mensagem_codificada, mapa_codificacao
```

```
def decodificar_mensagem(mensagem_codificada, mapa_codificacao):
    mensagem_original = ""

    # Inverte o dicionário para decodificação
    mapa_decodificacao = {v: k for k, v in mapa_codificacao.items()}

    for caractere in mensagem_codificada:
        if caractere in mapa_decodificacao:
            mensagem_original  += mapa_decodificacao[caractere]
        else:
            mensagem_original  += caractere

    return mensagem_original
```

In [76]:
```
# Teste
mensagem = "Segredo: Vamos ao bar às 18h!"

# Codifica a mensagem
mensagem_codificada, mapa = codificar_mensagem(mensagem)

# Mostra a mensagem codificada
print("Mensagem original:", mensagem)
print("Mensagem codificada:", mensagem_codificada)

# Decodifica a mensagem
mensagem_decodificada = decodificar_mensagem(mensagem_codificada, mapa)
print("Mensagem decodificada:", mensagem_decodificada)
```

```
Mensagem original: Segredo: Vamos ao bar às 18h!
Mensagem codificada: BqEUqaR: ZyLRH yR TyU XH 18A!
Mensagem decodificada: Segredo: Vamos ao bar às 18h!
```

In [ ]: