RAPPORT DU TP1 : CODE CESAR

Dans ce TP, on a vu 3 types de chiffrage qui sont :

CODE INVERSE :

Il s'agit de crypter un message par son écriture à l'envers. Le décrypter revient au même. Un seul programme sera utilisé en considérant qu'on est en mode cryptage ou décryptage.

Code :

```
def crypt(mess):
    traduction = ""
    i = len(mess) - 1
    while i > -1:
        traduction = traduction + mess[i]
        i = i - 1
    return traduction
```

Exemple :

Si mot = "HELLO" donc

```
m = crypt(mot)
print(m)
print(crypt(m))
```

Donne le résultat suivant :

OLLEH

HELLO

CODE CESAR:

- SIMPLE:

Pour crypter, on décale chaque lettre de b indices où b est la clé de chiffrement. Crypter le message avec la clé b sur un alphabet de taille n : $y \equiv x + b[n]$, y est le message obtenu.

Décrypter le message x avec la clé b sur un alphabet de taille n : $y \equiv x - b[n]$, y est le message obtenu.

La clé b est un nombre entier et une variable indique en quel mode on agit (cryptage ou décryptage).

- AFFINE :

Crypter le message x avec les clés a et b sur un alphabet de taille n : $y \equiv ax + b[n]$, y est le message obtenu.

Décrypter le message x avec les clés a' et b': $y \equiv a'x + b'[n]$, y est le message obtenu.

CODE :

```python
# code cesar pour (dé)chiffrer
def cesar(m, b, a=1):
    # si a et 26 ne sont pas premier entre eux: arrête de prog.
    if pgcd(a, 26) != 1:
        print("erreur, ", a, " et 26 ne sont premier entre eux")
        return -1
    elif a < 0:
        print("erreur,", a, "n'est pas un entier")
        return -1

    print("->mode cryptage<-") if b > 0 else print("->mode décryptage<-")
    global alph
    mot = ""
    for i in m:
        # (dé)chiffrer le 1er caractère du mot m
        nb = (int(alph[i]) * a + b) % 26 if b > 0 else ((int(alph[i]) + b) * invr(a, 26)) % 26
        # chercher le caractère correspondent au nombre trouvé, et l'ajouter au mot "mot" qu'on veut construire
        for car, val in alph.items():
            if int(val) == nb:
                mot += car
    if b >= 0:
        print("le mot chiffré est", mot, "\n")
    else:
        print("le mot déchiffré est", mot, "\n")
    return mot
```

Avec :

```python
alph = {"A": "00", "B": "01", "C": "02", "D": "03", "E": "04", "F": "05", "G": "06", "H": "07", "I": "08", "J": "09",
        "K": "10", "L": "11", "M": "12", "N": "13", "O": "14",
        "P": "15", "Q": "16", "R": "17", "S": "18", "T": "19", "U": "20", "V": "21", "W": "22", "X": "23", "Y": "24",
        "Z": "25"}


# trouver l'inverse de x en modulo mo
def invr(x, mo):
    i = 2
    # multiplier x par tous les nombres jusqu'à trouver i tel que x*i congru 1 modulo 26
    while (x * i) % mo != 1:
        i += 1
    return i


# trouver le pgcd de a et b
def pgcd(a, b):
    if b == 0:
        return a
    else:
        return pgcd(b, (a % b))
```

Exemple de César simple : si b=3

Texte clair = "HELLO" (7,4,11,11,14) → cryptage texte chiffré (10,7,14,14,17) "KHOOR" → décryptage
Texte clair = "HELLO"

Le code suivant :

```
b = 3
mot = "HELLO"

print('\nexemple de chiffrage et déchiffrage du mot "HELLO" avec un code cesar simple:\n')
m = cesar(mot, b)
cesar(m, b * -1)
```

Donne :

```
exemple de chiffrage et déchiffrage du mot "HELLO" avec un code cesar simple:

->mode cryptage<-
le mot chiffré est KHOOR

->mode décryptage<-
le mot déchiffré est HELLO
```

Exemple de César affine : si b=3 et a = 5

Texte clair = "HELLO" (7,4,11,11,14) → cryptage texte chiffré (12,23,6,6,21) " MXGGV" → décryptage
Texte clair = "HELLO"

Le code suivant :

```
a = 5
b = 3
mot = "HELLO"
print('\nexemple de chiffrage et déchiffrage du mot "HELLO" avec un code cesar affine:\n')
m = cesar("HELLO", b, a)
cesar(m, b * -1, a)
```

Donne :

```
exemple de chiffrage et déchiffrage du mot "HELLO" avec un code cesar affine:

->mode cryptage<-
le mot chiffré est MXGGV

->mode décryptage<-
le mot déchiffré est HELLO
```