Illiam
Pon-layus
NSI

J’ai compris rapidement le chiffre vigenere sur papier et avec des exemple mais le retrenscricre en python à était beaucoup plus compliqué ,j’ai relue ,relue mais leçon.
J’écrivais sur un brouillion toute les idée prométante et les testé ,pendant bien 2 h.

```python
alphabet = ['A', 'B', 'C', 'E', 'D', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', ''R'', 'S', 'T', 'U',
            'V', 'W', 'X', 'Y', 'Z']

def chiffrement(cle, message):
    index_cle = 0
    for i ,c in range(0,len(message-1,len(alphabet)):
       if (message[i]==alphabet[c]) :
            message[i] = alphabet[c]
```

Ce programme était une des idée qui me traversais l’esprit mais elle n’est pas aboutit car je savais pas du tout quoi mettre ,je chercher en tatonnant. Alord l’idée de ce programme était
ce que je chercher c’était que si l’index de message est égale à l’index de l’alphabet alors on affecte l’index de alphabet à l’index de
message .Mais ensuite j’était bloquée parce que les personne qui expliquer le chiffre
de vigenere ,il me parlé de modulo 26 mais je voyer pas comment faire dans python .Donc j’ai abandonner cette piste.

```python
def vigenere(cle, message):

 # La fonction d'encodage se base sur :
  # message : le message à chiffrer en  chaine de caractères
 # cle : la clé à utiliser en  chaine de caractères
  # Elle retourne le message chiffré en  chaine  caractères

    indice_cle = 0
    msg_code = ""

    for i in range(0, len(message)):
        if 'A' <= message[i] <= 'Z':
            msg_code += chr((((ord(message[i]) - ord('A')) + (ord(cle[indice_cle]) - ord('A'))) % 26) + ord('A'))

            indice_cle = (indice_cle + 1) % len(cle) #on incremente 1 et le divise par la longueur de la clé qui
        else:
            msg_code += message[i]  # on ajoute le tous les index de message au  message codé
    return msg_code # on retourne en programme le messsage codé
```

Vu que je ne voyer pas par où commencer j’ai demandé à un terminale(spé nsi) juste de m’aiguillé pour que je puisse me lancé et également mattéo sans qu’il me donne la réponse .
Donc maintennat que je me suis renseigné ,j’ai réussi à commencer enfin .
En premier lieu crée ma fonction avec en paramétre clé et message que l’on veut codé.
Je mes indcice_cle à 0 pour pouvoir l’incrementer et faire la division de la longeur de la chaine clé plus loin dans le programme.
Là je mes une boucle pour que les calcul ce fasse sur l’entiereté de la chaîne de caratères
ensuite je mes une condition si l’index de message est superieure au code asscii de A et inférieur au code asscii de Z :
alors msg_code est égale aux code asscii de l’index de message moins le code asscii de A plus le code asscii de cle avec index indice_cle moins le code asscii de A tous ça modulo de 26(maintenant j’ai compris monsieur herman m’a expliquer) plus code asscii de A tous cela en chaîne da caratères(chr)
else msg_code s’ajoute et s’affecte avec =message[i]

aprés on retourne le message codé comme ça quand on appelle le fonction ou qu'on l'affiche c'est celui-ci qui appaitra .

```
    indice_cle = 0
    msg_code = ""

    for i in range(0, len(message)):
        if 'A' <= message[i] <= 'Z':
            msg_code += chr((((ord(message[i]) - ord('A')) - (ord(cle[indice_cle]) - ord('A'))) % 26) + ord('A'))
            indice_cle = (indice_cle + 1) % len(cle)
        else:
            msg_code += message[i]
    return msg_code

rint(vigenere('PYTHON', 'BONJOUR A TOUS'))
rint(decode_vigenere('PYTHON', 'QMGQCHG Y MVIF'))
```

```
chr((((ord(message[i]) - ord('A')) - (ord(cle[indice_cle]))
```

Pour le décodage de vigenere on fait pratiquement pareille mais à la place de additionner la code assciide de cle [indice_cle] on le soustrait