```python
# By BEAUQUIER Quentin

# Partie 1
# 1.1

ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

# Q1.1
def position(letter):
    return ALPHABET.find(letter.upper())
# Q1.2
def decalage(letter, dec):
    if letter.upper() in ALPHABET:
        if letter.isupper():
            return ALPHABET[(position(letter.upper())+dec)%len(ALPHABET)]
        else:
            return ALPHABET[(position(letter.upper())+dec)%len(ALPHABET)].lower()

# Q1.3
def cryptage(text, dec):
    encoded = ""
    for letter in text:
        if letter.upper() in ALPHABET:
            encoded += decalage(letter, dec)
        else:
            encoded += letter
    return encoded

# Q1.4
myText = "Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascet

# decalage de 15
print(cryptage(myText, 15))

# 1.2
# Q2.1
"""
Le chiffrement de CÃ©sar fait un dÃ©calage constant sur chaque caractÃ¨re donc connaitre le dÃ©calage
d'un caractÃ¨re c'est connaitre le dÃ©calage de tous les caractÃ¨res.
"""

# Q2.2
def occurency_nb(text, x):
    return text.upper().count(x.upper())

# Q2.3
def most_frequently(text):
    maxValue = 0
    currentValue = 0
    maxLetter = ''
    for letter in ALPHABET:
        currentValue = occurency_nb(text, letter)
        if currentValue >= maxValue:
            maxValue = currentValue
            maxLetter = letter
    return maxLetter

# Q2.4
def decryptage(code):
    print("----------------------------------------")
    print("DÃ©codage d'un texte en franÃ§ais")
    print("----------------------------------------")

    frequentLetter = ['e', 'a', 's', 'i', 't', 'n', 'u', 'o']
    actualFrequentLetter = most_frequently(code)
    possibleDec = []
    for i in frequentLetter:
        if position(i) < position(actualFrequentLetter):
            possibleDec.append(26 - abs(position(i)-position(actualFrequentLetter)))
        else :
            possibleDec.append(abs(position(i)-position(actualFrequentLetter)))
    for dec in possibleDec:
        decoded = ""
        for letter in code:
            if letter.upper() in ALPHABET:
                decoded += decalage(letter, dec)
            else:
                decoded += letter
        print(decoded)
        print("----------------------------------------")



print("----------------")

myText2 = "Dans un royaume lointain, un jeune apprenti sorcier dÃ©couvrit un livre ancien contenant des sorts interdits, dÃ©clenchant ainsi une sÃ©rie d'Ã©vÃ©nements magiques imprÃ©vis

import random
n = random.randint(0, 25)
print(n)
coded = cryptage(myText2, n)

decryptage(coded)

# Partie 2
# Q3.1

def cryptage_substitution(text, subs):
    encoded = ""

    for letter in text:
        if letter.upper() in ALPHABET:
            encoded += subs[position(letter)]
        else :
            encoded += letter
    return encoded

print(cryptage_substitution("salut", "xnyahpogzqwbtfsflrcvmiuekjdi"))

# Q3.2

def decryptage_substitution(code, subs):
    decoded = ""

    for letter in code:
        if letter.upper() in ALPHABET:
            if letter.isupper():
                decoded += ALPHABET[subs.find(letter)]
            else:
                decoded += ALPHABET[subs.find(letter)].lower()
```

```
        else :
            decoded += letter
    return decoded

print(decryptage_substitution("cxbmv", "xnyahpogzqwbtfsflrcvmiuekjdi"))

# Partie 3
# Q4.1

def vigenere_chiffrement(text, cle):
    encoded = ""
    decCle = []
    for i in cle:
        decCle.append(position(i))
    for i in range(0, len(text)):
        if text[i].upper() in ALPHABET:
            encoded += decalage(text[i], decCle[i%len(decCle)])
        else :
            encoded += text[i]
    return encoded

def vigenere_dechiffrement(text, cle):
    decoded = ""
    decCle = []
    for i in cle:
        decCle.append(position(i))
    for i in range(0, len(text)):
        if text[i].upper() in ALPHABET:
            decoded += decalage(text[i], -decCle[i%len(decCle)])
        else :
            decoded += text[i]
    return decoded

print(vigenere_chiffrement("Ceci est un test", "poule"))

print(vigenere_dechiffrement("Rswt tgn yc npwi", "poule"))
```