

```

alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
#Q1
#c : lettre
def rang(c) :
    return ord(c)-65 #65 = ord('A')

#r : rang
def lettre(r):
    return chr(r+65)

#Q2
def vigenere(m,cle):
    """
    m : message à crypter
    cle : clé du cryptage
    mc : message crypté
    """
    #alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    mc = "" #message codé
    i = 0 #compteur de parcours de la clé
    for c in m :
        if c in alphabet:
            # i % len(cle) = position dans la clé
            mc += lettre(rang(c) + rang(cle[i % len(cle)]))
            i += 1
        else:#si c n'est pas dans ['A'..'Z']
            mc += c #retourne le même caractère c
    return mc

#tests
print(rang('B'))
print(lettre(1))
mc =vigenere("CETTE PHRASE DE VEUT RIEN DIRE", "DBFC")
print(mc)

#Q3
def devigenere(mc,cle):
    """
    m : message d'origine
    cle : clé du cryptage
    mc : message à decrypté
    """
    m = ""
    i = 0 #compteur de parcours de la clé
    for c in mc :
        if c in alphabet:
            # i % len(cle) = position dans la clé
            r = rang(c) - rang(cle[i % len(cle)])
            m += lettre(r % len(alphabet) )
            i += 1
        else:#si c n'est pas dans ['A'..'Z']
            m += c #retourne le même caractère c
    return m

```

```
mc = "FFYVH QMTDTJ FH WJWW SNGQ ENTH"  
cle = "DBFC"  
m = devigenere(mc, cle)  
print(m)
```