```
###############
###FONCTIONS###
###############

def morse_encodage(txt): ##définition de la fonction
    txt=txt.upper() ##pour s'accorder avec le dictionnaire
    txt=[*txt] ##création d'une liste qui sépare tous les caractères de txt
    txt_final=""
    x=0 ##compteur
    for i in range(len(txt)):
        if txt[i] not in caracteres_morse.keys():
            return "Erreur. Le caractère ",txt[i]," ne peut pas être traduit en
morse."
        elif all(c in ".- " for c in txt):
            return "Erreur. Le texte que vous avez entré est déjà en morse."
    for i in range(len(txt)):
        if txt[i]!=" ":
            txt_final+=caracteres_morse[txt[x]]+" " ##on ajoute " " à la fin pour
séparer les caratères
        else :
            txt[i]=" " ##on remplace un espace par " "
            txt_final+=" " ##on obtient "  " pour séparer deux mots
        x+=1
    return txt_final


def morse_decodage(txt):  ##définition de la fontion
    txt+=" " ##pour signifier la fin
    cara_morse=""
    txt_final=""
    x=0 ##compteur
    for i in range(len(txt)): ##vérification des caractères
        if txt[i] not in caracteres_morse.values():
            return("Erreur. Le caractère ",txt[i]," n'existe pas en morse.")
    for i in range(len(txt)): ##on extrait le caractère correspondant au morse
        if txt[i]!=" ":
            cara_morse+=txt[i]
            x=0 ##pour compter les espaces (" ")
        else:
            x+=1
            if x==2: ##nouveau mot ("  ")
                txt_final+=" "
            else: ##recherche inversée dans le dictionnaire
                j=list(caracteres_morse.values()) ##on extrait toutes les valeurs
du dictionnaire
                j=j.index(cara_morse) ##trouver la position de cara_morse[i] dans
les valeurs dictionnaire
                txt_final+=list(caracteres_morse.keys())[j] ##on extrait la clef
correspondant à la valeur (j)
                cara_morse="" ##on réinitialise la variable
    return txt_final


def vigenere(quoi,txt,clef): ##définition de la fonction
    ##définition des variables
    clef_ascii=[]
    txt_ascii=[]
    txt_final=""
    maj_min_sym=[] ##pour différencier les majuscules, minuscules et
symboles/chiffres
    t=1 ##devient -1 si on cherche à décoder le texte
    j=0 ##compteur
    if clef=="": ##vérification qu'il n'y a pas d'erreur
        return "Erreur de clef. Veuillez recommencer."
    if quoi=="décoder": ##inversion pour faire le décodage
        t=-1
    clef=clef.lower()
```

```
    for i in clef: ##transformation de la clef en un tableau des caractères ascii
correspondants
        clef_ascii.append(t*(ord(i)-97)) ##-97 pour que a=0, b=1, etc
    while len(clef_ascii)<len(txt): ##assemblage de la clef pour avoir assez de
caractères
        clef_ascii+=clef_ascii
    clef_ascii=clef_ascii[:len(txt)]
    for i in txt: ##transformation du texte à crypter en ascii
        if 97<=ord(i)<=122: ##minuscules
            txt_ascii.append(ord(i)-97) ##-97 idem
            maj_min_sym.append("m")
        elif 65<=ord(i)<=90: ##majuscules
            txt_ascii.append(ord(i)-65) ##-65 pour que A=0, B=1, etc
            maj_min_sym.append("M")
        else: ##symboles/chiffres
            txt_ascii.append(ord(i)) ##on garde le même
            maj_min_sym.append("")
    for i in txt_ascii: ##encodage de chaque caractère selon son type
        if maj_min_sym[j]=="m": ##minuscules
            x=(i+clef_ascii[j])%26 ##(texte[i]+clef[i])%26 pour encoder, (texte[i]-
clef[i])%26 pour décoder
            txt_final+=chr(x+97)
        elif maj_min_sym[j]=="M": ##majuscules
            x=(i+clef_ascii[j])%26
            txt_final+=chr(x+65)
        else: ##symboles/chiffres
            txt_final+=chr(i)
        j+=1 ##compteur
    return (txt_final)


def cesar(quoi,txt,decalage,direction): ##définition de la fontion
    ##définition des variables
    txt_final=""
    txt_ascii=[]
    type_caractere=[] ##pour pouvoir distinguer le type de caractère (lettre
majuscule ou minuscule, chiffre, symbole)
    if direction=="gauche" : ##inversion du décalage si on veut décoder ou s'il se
fait vers la gauche
        decalage*=-1
    if quoi=="décoder" :
        decalage*=-1
    for i in txt: ##transformation du texte à crypter en un tableau des ascii
correspondant à chaque caractère
        txt_ascii.append(ord(i))
    for i in range(len(txt_ascii)): ##encodage de chaque caractère selon son type
        if 97<=txt_ascii[i]<=122: ##minuscules
            txt_ascii[i]=(txt_ascii[i]+decalage-97)%26 ##-97 pour retrouver une
lettre minuscule ##%26 pour retrouver une lettre de l'alphabet
            type_caractere.append("m") ##on enregistre que c'est une minuscule
        elif 65<=txt_ascii[i]<=90: ##majuscules
            txt_ascii[i]=(txt_ascii[i]+decalage-65)%26 ##-65 idem majuscule ##idem
            type_caractere.append("M") ##idem majuscule
        elif 48<=txt_ascii[i]<=57: ##chiffres
            txt_ascii[i]=(txt_ascii[i]+decalage-48)%10 ##-48 pour retrouver un
chiffre ##idem chiffre
            type_caractere.append("c") ##idem chiffre
        else: ##symboles
            type_caractere.append("s") ##idem symbole
    for i in range(len(txt_ascii)): ##transformation de l'ascii crypté en texte
        if type_caractere[i]=="m": ##minuscules
            txt_final+=chr(txt_ascii[i]+97) ##+97 pour retrouver une minuscule
        elif type_caractere[i]=="M": ##majuscules
            txt_final+=chr(txt_ascii[i]+65) ##+65 idem majuscule
        elif type_caractere[i]=="c":
            txt_final+=chr(txt_ascii[i]+48) ##+48 idem chiffre
        elif type_caractere[i]=="s": ##symboles
            txt_final+=chr(txt_ascii[i])
```

```
    return(txt_final)



##########
###CODE###
##########

##création du dictionnaire pour le morse##
caracteres_morse= { "A":".-", "B":"-...", "C":"-.-.", "D":"-..", "E":".", "F":"..-
.", "G":"--.", "H":"....", "I":"..", "J":".---", "K":"-.-", "L":".-..", "M":"--",
"N":"-.", "O":"---", "P":".--.", "Q":"--.-", "R":".-.", "S":"...", "T":"-",
"U":"..-", "V":"...-", "W":".--", "X":"-..-", "Y":"-.--", "Z":"--..", "1":".----",
"2":"..---", "3":"...--", "4":"....-", "5":".....", "6":"-....", "7":"--...",
"8":"---..", "9":"----.", "0":"-----", ".":".-.-.-", "?":"..--..", "-":"-....-",
"'":".----.", " ":" ", ",":"--..--"}

redo="oui" ##pour pouvoir rejouer le programme

print("Bienvenue.")
print("Ici, vous pourrez encoder ou décoder en utilisant le chiffre de César, celui
de Vignère ou encore le code Morse.")

while redo=="oui":
    quel=input("Lequel voulez-vous utiliser ? (César/Vigenère/Morse): ")
    if quel!="César" and quel !="Vigenère" and quel!="Morse": ##vérification de
l'entrée
        print("")
        print("Erreur. Vérifiez l'orthographe.")
        print("")
        continue ##retour au début
    quoi=input("Que voulez-vous faire ? (encoder/décoder): ")
    if quoi!="encoder" and quoi!="décoder": ##vérification de l'entrée
        print("")
        print("Erreur. Vérifiez l'orthographe.")
        print("")
        continue ##retour au début
    txt=input("Entrez le texte: ")
    if quel=="Morse":
        if quoi=="encoder":
            print("")
            print(morse_encodage(txt)) ##éxécution de la fonction
        elif quel=="décoder":
            print("")
            print(morse_decodage(txt)) ##éxécution de la fontion
    elif quel=="César":
        decalage=int(input("Entrez la valeur du décalage à l'encodage: "))
        direction=input("Entrez la direction (droite/gauche): ")
        if direction!="droite" and direction!="gauche": ##vérification de l'entrée
            print("")
            print("Erreur. Vérifiez l'orthographe.")
            print("")
            continue ##retour au début
        print("")
        print(cesar(quoi,txt,decalage,direction)) ##éxécution de la fonction
    elif quel=="Vigenère":
        clef=input("Entrez la clef de chiffrage (sous forme de texte): ")
        print("")
        print(vigenere(quoi,txt,clef)) ##éxécution de la fonction
    print("")
    redo=input("Voulez-vous recommencer ? (oui/non): ")
    if redo=="non":
        print("Au revoir.")


###########
###TESTS###
###########
```

```
#####Vigenère###
##
##if vigenere("encoder","Hello World !","clef")=="Jppqq Attwh !":
##    print("Test encodage Vigenère OK")
##else:
##    print("Test encodage Vigenère pas OK")
##
##if vigenere("décoder","Jppqq Attwh !","clef")=="Hello World !":
##    print("Test décodage Vigenère OK")
##else:
##    print("Test décodage Vigenère pas OK :", vigenere("décoder","Jppqq Attwh
!","clef"))
##
##
#####César###
##if cesar("encoder","Hello World !",147,"droite")=="Yvccf Nficu !":
##    print("Test encodage César OK")
##else:
##    print("Test encodage César pas OK", cesar("encoder","Hello World
!",147,"droite"))
##
##if cesar("décoder","Yvccf Nficu !",147,"droite")=="Hello World !":
##    print("Test décodage César OK")
##else:
##    print("Test décodage César pas OK :", cesar("décoder","Yvccf Nficu
!",147,"droite"))
##
##
#####Morse###
##caracteres_morse= { "A":".-", "B":"-...", "C":"-.-.", "D":"-..", "E":".",
"F":"..-.", "G":"--.", "H":"....", "I":"..", "J":".---", "K":"-.-", "L":".-..",
"M":"--", "N":"-.", "O":"---", "P":".--.", "Q":"--.-", "R":".-.", "S":"...", "T":"-
", "U":"..-", "V":"...-", "W":".--", "X":"-..-", "Y":"-.--", "Z":"--..", "1":".----
", "2":"..---", "3":"...--", "4":"....-", "5":".....", "6":"-....", "7":"--...",
"8":"---..", "9":"----.", "0":"-----", ".":".-.-.-", "?":"..--..", "-":"-....-",
"'":".----.", " ":" ", ",":"--..--"}
##if morse_encodage("Hello World")==".... . .-.. .-.. ---  .-- --- .-. .-.. -.. ":
##    print("Test encodage Morse OK")
##else:
##    print("Test encodage Morse pas OK :", morse_encodage("Hello World"))
##
##if morse_decodage(".... . .-.. .-.. ---  .-- --- .-. .-.. -.. ")=="HELLO WORLD ":
##    print("Test décodage Morse OK")
##else:
##    print("Test décodage Morse pas OK :", morse_decodage(".... . .-.. .-.. ---
.-- --- .-. .-.. -.. "))
##
```