*On veut chiffrer la phrase « CRYPTONS CETTE PHRASE AVEC LA MACHINE ENIGMA ».*

*Tout d’abord, définissons l’alphabet :*

alphabet\_1="ABCDEFGHIJKLMNOPQRSTUVWXYZ"

*Posons aussi :*

phrase= "CRYPTONSCETTEPHRASEAVECLAMACHINEENIGMA"

*Ensuite, comme c’est plus simple de travailler avec des nombres plutôt que des chiffres :*

def conversion\_nombre(alphabet,lettre):

"""On change une lettre en un nombre : A->0, B->1, C->2, D->3, … , Z->25"""

for i in range(len(alphabet)):

if lettre==alphabet[i]: # on cherche parmi les éléments de « alphabet » lequel correspond à la lettre donnée en argument.

return i #on retourne la position (- 1…) de la lettre dans « alphabet »

def conversion\_nombre\_liste(alphabet,texte):

"""On change un texte en une liste de nombres avec la fonction conversion\_nombre écrite précédemment. Par exemple, conversion\_nombre\_liste(alphabet,"TEXTE") renvoie [19,4,23,19,4]"""

liste\_de\_nombres=[]

for i in texte:

liste\_de\_nombres.append(conversion\_nombre(alphabet,i))

return liste\_de\_nombres

*En fait :*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *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* |
| *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* | *23* | *24* | *25* |

*conversion\_nombre\_liste(alphabet\_1,"CRYPTONSCETTEPHRASEAVECLAMACHINEENIGMA") renvoie ainsi [2, 17, 24, 15, 19, 14, 13, 18, 2, 4, 19, 19, 4, 15, 7, 17, 0, 18, 4, 0, 21, 4, 2, 11, 0, 12, 0, 2, 7, 8, 13, 4, 4, 13, 8, 6, 12, 0]*

*De même, profitons-en pour définir une fonction qui a un nombre entre 0 et 25 renvoie une lettre de l’alphabet :*

def conversion\_lettre(alphabet,nombre):

return alphabet[nombre%len(alphabet)]

def conversion\_lettre\_liste(alphabet,liste):

newliste=[]

for i in liste:

newliste.append(conversion\_lettre(alphabet,i))

return newliste

*Ensuite arrive le tableau de connexion. Il se présente sous la forme d’une liste de liste à deux éléments. Par exemple, pour chiffrer notre « phrase », prenons :*

tableau\_de\_connexion\_numero\_1=[[

""

""

""

""

""

""

def substitution(alphabet\_de\_depart,alphabet\_de\_cryptage,texte): #bon ca ca marche!!!

n=len(alphabet\_de\_depart)

m=len(alphabet\_de\_cryptage)

newtext=[] #ce sera le texte crypté

if n!=m:

return "Y a un problème dans la longueur des deux alphabets"

else:

for i in range(len(texte)):

k=texte[i] # on prend la i-ième lettre dans texte

for j in range(n):

if k==alphabet\_de\_depart[j]:

newtext.append(alphabet\_de\_cryptage[j])

return newtext

CONVERSION LETTRE-NOMBRE :

def nouvel\_alphabet(alphabet,liste\_de\_deux\_elements):

new\_alphabet=conversion\_nombre\_liste(alphabet,alphabet)

for substitution in liste\_de\_deux\_elements:

for chiffre in conversion\_nombre\_liste(alphabet,alphabet):

if chiffre==substitution[0]:

new\_alphabet[substitution[0]]=substitution[1]

new\_alphabet[substitution[1]]=substitution[0]

return new\_alphabet

PASSAGE ROTOR :

def passage\_rotor(rotor,lettre,indice\_de\_la\_lettre,place\_du\_rotor,nombre\_de\_rotors):

r=len(rotor)

return (lettre+rotor[(lettre+(indice\_de\_la\_lettre-1)//(r\*\*(nombre\_de\_rotors-place\_du\_rotor)))%r])%r

def passage\_inverse\_rotor(rotor,lettre,indice\_de\_la\_lettre,place\_du\_rotor,nombre\_de\_rotors):

for i in range(len(rotor)):

if passage\_rotor(rotor,i+1,indice\_de\_la\_lettre,place\_du\_rotor,nombre\_de\_rotors)==lettre:

return i+1

def passage\_reflecteur(reflecteur,lettre):

def machineenigma(tableaudeconnexion,listedesrotors,reflecteur,texte):

n=len(texte)

m=len(listedesrotors)

newtext=[]

for i in range(n):

nouveautexte=substitution(conversionnombreliste(alphabet),newalphabet(tableaudeconnexion),conversionnombreliste(texte))

for j in range(m):

nouveautexte[i]=passagerotor(listedesrotors[j],nouveautexte[i],i+1,j+1,m)

print(["a l'aller",i,conversionlettre(nouveautexte[i])])

nouveautexte[i]=passagereflecteur(reflecteur,nouveautexte[i])

print(conversionlettre(nouveautexte[i]))

for k in range(m):

nouveautexte[i]=passageinverserotor(listedesrotors[m-k-1],nouveautexte[i],i+1,m-k,m)

print(["au retour",i,conversionlettre(nouveautexte[i])])

newtext.append(nouveautexte[i])

return conversionlettreliste(newtext)

def machineenigma2(tableaudeconnexion,listedesrotors,reflecteur,texte):

n=len(texte)

m=len(listedesrotors)

newtext=[]

for i in range(n):

nouveautexte=substitution(conversionnombreliste(alphabet),newalphabet(tableaudeconnexion),conversionnombreliste(texte))

for j in range(m):

nouveautexte[i]=passagerotor(listedesrotors[j],nouveautexte[i],i+1,j+1,m)

nouveautexte[i]=passagereflecteur(reflecteur,nouveautexte[i])

for k in range(m):

nouveautexte[i]=passageinverserotor(listedesrotors[m-k-1],nouveautexte[i],i+1,m-k,m)

newtext.append(nouveautexte[i])

return conversionlettreliste(newtext)

rotor1=[3,3,24,5,7,22,1,3,20,9,4,5,5,6,24,5,23,7,17,22,1,1,10,2,25,6]

rotor2=[9,22,6,9,21,19,1,6,12,18,12,3,14,19,17,4,25,12,19,9,16,23,8,19,23,18]