CRYPTOGRAPHIE / CHIFFREMENT

La fonction **`ord`**`(char)` renvoie le code ASCII d'un caractère. La fonction **`chr`**`(num)` renvoie le caractère ASCII correspondant (`num` doit être compris entre 0 et 255).

On utilise les chaînes de caractères comme des tableaux mais on en ne peut pas modifier le contenu d'une chaîne existante.

Les noms de fonctions doivent être conservés. Le fichier à rendre doit contenir un algorithme des fonctions et des tests. Le programme doit être commenté.

I. TECHNIQUES HISTORIQUES

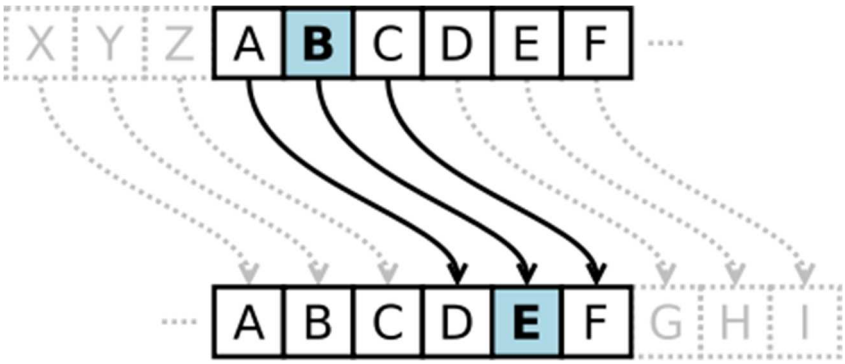
1) CODE DE CESAR

Le **chiffrement par décalage**[1], aussi connu comme le **code de César**, est une méthode de chiffrement très simple utilisée par Jules César dans ses correspondances secrètes.

Le texte chiffré s'obtient en remplaçant chaque lettre du texte clair original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet. Pour les dernières lettres (dans le cas d'un décalage à droite), on reprend au début. Par exemple avec un décalage de 3 vers la droite, `A` est remplacé par `D`, `B` devient `E`, et ainsi jusqu'à `W` qui devient `Z`, puis `X` devient `A` etc. Il s'agit d'une permutation circulaire de l'alphabet.

La longueur du décalage, 3 dans l'exemple évoqué, constitue la *clé* du chiffrement qu'il suffit de transmettre au destinataire pour que celui-ci puisse déchiffrer le message.

A titre d'exemple, dans le film *L'odyssée de l'espace*, l'ordinateur porte le nom de *HAL*. Ce surnom est en fait *IBM* décalé de 1 position vers la gauche...



Question I-1 : Coder le mot « CRYPTOGRAPHIE » avec la clé 4.

Question I_2 : Écrire une fonction en Python qui applique le code de César sur une lettre.

```
def Cesar_lettre (lettre,cle) :
```

[1] https://fr.wikipedia.org/wiki/Chiffrement_par_décallage

La lettre à coder doit être en majuscule (Codes ASCII 65 à 90). Si elle est en minuscule, elle doit être convertie en majuscule (Cf. fonction `upper`). Les caractères qui ne sont pas des lettres ne sont pas modifiés.

REMARQUE : LA CLÉ PEUT ÊTRE NÉGATIVE.

EXEMPLES :

```
In [43]: Cesar_lettre('a',3)
Out[43]: 'D'

In [44]: Cesar_lettre('Z',3)
Out[44]: 'C'
```

```
In [40]: Cesar_lettre(' ',3)
Out[40]: ' '

In [41]: Cesar_lettre('>',3)
Out[41]: '>'
```

Question I_3 : Écrire une fonction en Python qui applique le code de César sur un mot entier en majuscules (les minuscules sont transformées en majuscules).

```
def Cesar (texte,cle) :
```

REMARQUE : Seules les lettres sont modifiées. Les espaces et les signes de ponctuation sont inchangés.

EXEMPLES :

```
In [59]: Cesar('abc WXY',3)
Out[59]: 'DEF ZAB'
```

Question I_4 : Écrire une fonction en Python qui décode du texte codé en utilisant le code de César. Le texte à coder doit être en majuscule sinon la fonction renvois une erreur.

```
def Decode_Cesar (texte,cle) :
```

```
In [78]: Decode_Cesar("DEF ZAB",3)
Out[78]: 'ABC WXY'
```

Question I_5 : Déterminer une méthode simple (mais pas forcément fiable à 100%) pour déterminer la clé de César à partir d'un texte assez long. Écrire une fonction qui décrypte un texte (assez long) codé.

```
def Decode_Cesar_v2 (texte) :
```

APPLICATION : Décoder informatiquement les textes suivants :

« RM NIQA AWCDMVB KM ZMDM MBZIVOM MB XMVMBZIVB L'CVM NMUUM QVKWVVCM MB YCM R'IQUM MB YCQ U'IQUM MB YCQ V'MAB KPIYCM NWQA VQ BWCB I NIQB TI UMUM VQ BWCB I NIQB CVM ICBZM MB U'IQUM MB UM KWUXZMVL.»

« MZFAZ HAKX Z'MDDUHMUF BME À PADYUD. UX MXXGYM. EAZ VML YMDCGMUF YUZGUF HUZSF. UX BAGEEM GZ BDARAZP EAGBUD, E'MEEUF PMZE EAZ XUF, E'MBBGKMZF EGD EAZ BAXAOTAZ. UX BDUF GZ DAYMZ, UX X'AGHDUF, UX XGF; YMUE UX Z'K EMUEUEEMUF CG'GZ UYNDASXUA OAZRGE, UX NGFMUF À FAGF UZEFMZF EGD GZ YAF PAZF UX USZADMUF XM EUSZURUOMFUAZ. QJFDMUF PQ XM PUEBMDUFUAZ ».

2) CODAGE PAR SUBSTITUTION MONOSYLLABIQUE

Ce codage consiste à remplacer chaque lettre par une lettre différente. Par exemple :

Alphabet clair	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alphabet de substitution	NBAJYFOWLZMPXIKUVCDEGRQSTH

Question I_6 : Coder le mot « CRYPTOGRAPHIE » avec ce code.

Question I_7 : Combien existent-t-il de clés (alphabets de substitution) possibles ?

Question I_8 : Écrire une fonction Python qui crée aléatoirement un alphabet de substitution (autrement dit, il suffit de mélanger aléatoirement les 26 lettres de l'alphabet). Le résultat est donc un tableau à 26 cases.

```
def alphabet_substitution () :
```

Question I_9 : Écrire une fonction qui code et une fonction qui décode du texte à partir d'un alphabet de substitution donné.

```
def Code_subst (texte,alphabet) :
def Decode_subst (texte,alphabet) :
```

EXEMPLES :

```
In [86]: Code_subst('AB cd:E',alphabet)
Out[86]: 'NB AJ:Y'

In [27]: Decode_subst('NB AJ:t',alphabet)
Out[27]: 'AB CD:t'
```

Pour le codage, les lettres sont transformées en majuscules. Les autres symboles (y compris l'espace) ne sont pas modifiés. Pour le décodage, seules les lettres majuscules sont décodées.

3) SYSTÈME DE VIGENERE

L'idée de Vigenère est d'utiliser un chiffre de César, mais où le décalage utilisé change de lettres en lettres. Pour cela, on utilise une table composée de 26 alphabets, écrits dans l'ordre, mais décalés de ligne en ligne d'un caractère. On obtient la table de Vigenère.

	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Pour coder un message, on choisit une clé qui sera un mot de longueur arbitraire. On écrit ensuite cette clé sous le message à coder, en la répétant aussi souvent que nécessaire pour que sous chaque lettre du message à coder, on trouve une lettre de la clé. Pour coder, on regarde dans le tableau l'intersection de la ligne de la lettre à coder avec la colonne de la lettre de la clé.

EXEMPLE : On veut coder le texte « LE CODAGE DE VIGENERE » avec la clé « CRYPTO ».

On écrit le texte à coder, et dessous, la clé, répétée autant de fois que nécessaire.

L	E	C	O	D	A	G	E	D	E	V	I	G	E	N	E	R	E
C	R	Y	P	T	O	C	R	Y	P	T	O	C	R	Y	P	T	O

Pour coder la 1ère lettre, on regarde l'intersection de la ligne « L » et de la colonne « C ». Ce qui donne « N ». Les 1ères lettres du codage sont :

N	V	A	D	W	O	I	V	B										

Question I_10 : Continuer le codage ci-dessus. Expliquer comment on décode le texte. Décoder le texte suivant avec la même clé pour connaître l'auteur et le titre du poème de la question I-5 ?

« XVPATWPV » « EVPTOSHRKXEWGI »

Question I_11 : À quelle condition, ce système de codage est-il fiable ?

Question I_12 : Écrire une fonction qui crée la table de Vigenère (le résultat est donc un tableau à 26 lignes et 26 colonnes). CONSEIL : utiliser les codes ASCII.

```
def Table_Vigenere () :
```

INDICATION : Pour simplifier, on remplace les lettres par des chiffres ('A'→0, 'B' → 1,…). Pour coder ou décoder, on utilisera les codes ASCII.

Question I_13 : Écrire des fonctions qui codent et décodent en utilisant le système de Vigenère.

```
def Code_Vigenere (texte,cle,table) :
def Decode_Vigenere (texte,cle,table) :
```

REMARQUE : Pour simplifier, on supprime les espaces du texte.

EXEMPLE :

```
In [29]: Code_Vigenere("LE CODAGE","CRYPTO",T)
Out[29]: 'NVADWOIV'
```

4- LE CARRE DE POLYBE

a) Version simple

Le carré de Polybe est un procédé de chiffrement par substitution. Il est basé sur le carré de 25 cases suivants contenant toutes les lettres de l'alphabet (sauf W) :

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	J
3	K	L	M	N	O
4	P	Q	R	S	T
5	U	V	X	Y	Z

Chaque lettre est représentée par ses coordonnées (ligne et colonne). Par exemple, la lettre F correspond à 21, la lettre R à 43.

La lettre W n'apparait pas dans l'alphabet. Elle sera remplacée par la lettre V.

Question I_14 : Écrire une fonction de codage et de décodage utilisant ce système. Pour le codage, les lettres sont systématiquement transformées en majuscules. Les autres caractères ne sont pas pris en compte. La lettre W est remplacée par la lettre V. La fonction de codage renvoie une liste de nombres à 2 chiffres.

```
def Code_Polybe_v1 (texte) :
def Decode_Polybe_v1 (liste) :
```

EXEMPLES :

```
In [95]: Code_Polybe_v1("Spyware")
Out[95]: [44, 41, 54, 52, 11, 43, 15]

In [1]: Decode_Polybe_v1([23, 35, 32, 32, 54, 52, 35, 35, 14, 24, 15, 34, 44])
```

b) Version améliorée

Pour améliorer le système, on va utiliser une clé privée (mot de passe). Par exemple, on choisit le mot « CRYPTOGRAPHIE » comme clé. On commence par supprimer les lettres en doublon. On obtient donc « CRYPTOGAHIE ». On remplit le tableau de Polybe avec ces lettres et on complète avec le reste de l'alphabet.

	1	2	3	4	5
1	C	R	Y	P	T
2	O	G	A	H	I
3	E	B	D	F	J
4	K	L	M	N	Q
5	S	U	V	X	Z

Question I_15 : Écrire une fonction qui crée ce tableau, et une fonction de codage et de décodage. On rempliera le tableau avec les nombres de 0 à 24 (A→0, B→1, …, Z-→25).

```
def Tableau_Polybe (cle) :
def Code_Polybe_v2 (texte, cle) :
def Decode_Polybe_v2 (liste, cle) :
```

EXEMPLES :

```
In [42]: Tableau_Polybe("CRYPTOGRAPHIE")
Out[42]:
array([[ 2.,  17.,  24.,  15.,  19.],
       [ 14.,  6.,  0.,  7.,  8.],
       [ 4.,  1.,  3.,  5.,  9.],
       [ 10.,  11.,  12.,  13.,  16.],
       [ 18.,  20.,  21.,  23.,  25.]])

In [94]: Code_Polybe_v2("SWITCHEZ","cryptographie")
Out[94]: [51, 53, 25, 15, 11, 24, 31, 55]

In [45]: Decode_Polybe_V2([51,53,25,15,11,24,31,55],"Cryptographie")
Out[45]: 'SVITCHEZ'
```

REMARQUE : Le plus simple est de supprimer le 22 dans le tableau de Polybe (correspondant à la lettre « W »).