



TP6 : Chaînes de caractères (Compléments)



Exercice 1 : Transcription en braille.

Louis Braille, est l'inventeur du système d'écriture tactile à points saillants, à l'usage des personnes aveugles ou fortement malvoyantes.

En braille standard :

- Un caractère est représenté par six points numérotés de 1 à 6 et disposés comme le montre la figure 1 ;
- Un point peut être saillant (en relief), ici marqué par les points en noir ;
- Le nombre et la disposition des points en relief définissent un caractère ;

Par la suite, on s'intéressera uniquement à la représentation des 59 caractères ASCII comme le montre la figure 2 (entre le caractère d'espace et la lettre Z majuscule).

| Représentation braille (figure 1) | Code ASCII réduit (figure 2) |
|--|--|
| 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 . , ; : / ? ! @ # + - * " " ' < > () capital _ and letter number | 32 espace 64 @ 33 ! 65 A 34 " 66 B 35 # 67 C 36 \$ 68 D 37 % 69 E 38 & 70 F 39 ' 71 G 40 (72 H 41) 73 I 42 * 74 J 43 + 75 K 44 , 76 L 45 - 77 M 46 . 78 N 47 / 79 O 48 0 80 P 49 1 81 Q 50 2 82 R 51 3 83 S 52 4 84 T 53 5 85 U 54 6 86 V 55 7 87 W 56 8 88 X 57 9 89 Y 58 : 90 Z 59 ; 60 < 61 = 62 > 63 ? |

Écrire la fonction « affichageBraille » afin de représenter un texte saisi par une personne en braille (compléter le fichier `braille.py`).

Exemple de rendu :

vousre texte ?touchard



Pour les exercices 2, 3 et 4, privilégier une structure du type « **for** indice **in** range() : » en réfléchissant bien aux paramètres de la boucle « pour ».

Exercice 2 : Insertion de caractères.

Écrire un script python qui recopie (dans une nouvelle variable) une chaîne saisie par une personne, en insérant des astérisques entre tous les caractères.

Exemple : « Touchard » devra devenir « T*u*c*h*a*r*d »

Fonction à compléter : `def insertionTexte(texte): #retourne une chaine de caractères`

Exercice 3 : Inversion de mots

Écrire un script python qui recopie (dans une nouvelle variable) une chaîne saisie par une personne) en l'inversant.

Exemple : « zorglub » deviendra « bulgroz ».

Fonction à compléter : `def inversionMot(texte): #retourne une chaine de caractères`

Exercice 4 : Palindrome

Écrire un script python qui détermine si une chaîne de caractères donnée est un palindrome (c'est-à-dire une chaîne qui peut se lire indifféremment dans les deux sens).

Exemple : « radar », « S.O.S. » ou « LAVAL ».

Fonction à compléter : `def palindrome(texte): #retourne un booléen True ou False`

Résultats affichés
dans la console

```

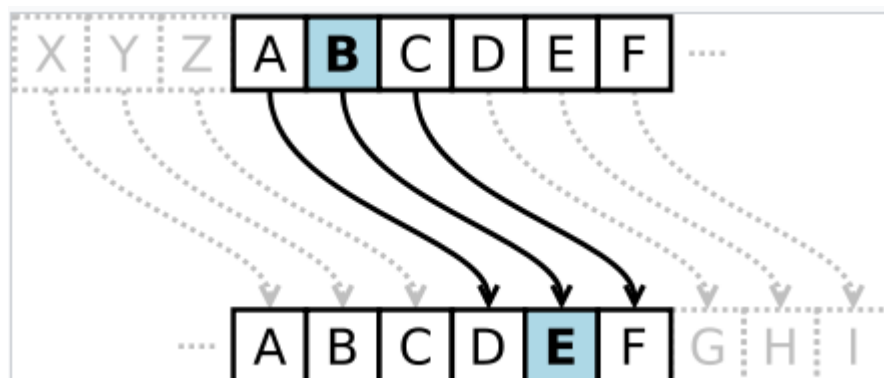
votre texte ?radar  votre texte ?toto
True                False

```

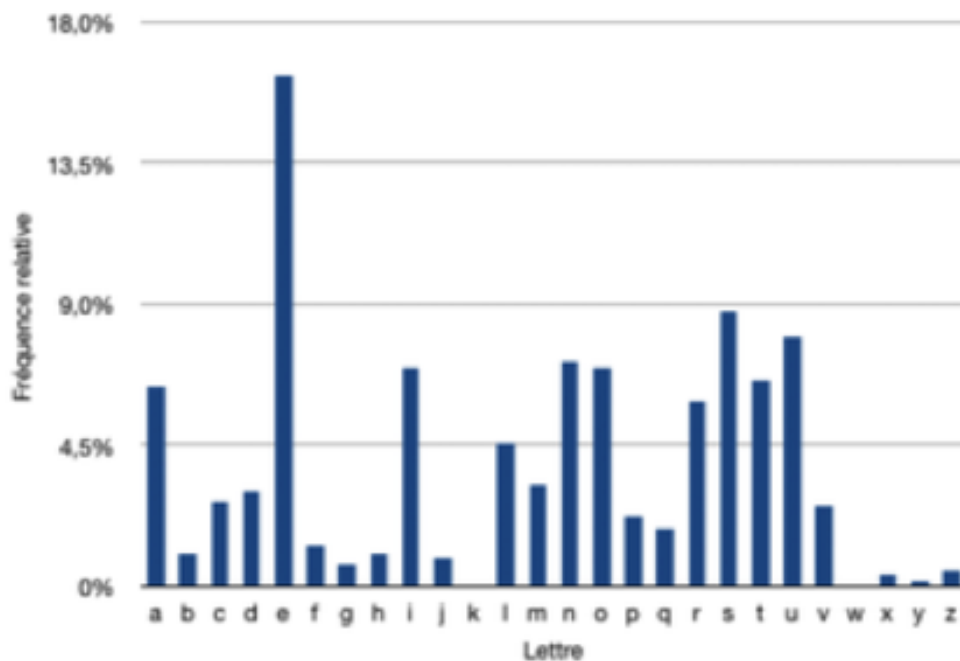
Exercice 5 : Codage de César ⇒ [Chiffrement par décalage](#)

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.

Exemple : Dans cet exemple, il y a une distance de 3 caractères, donc B devient E dans le texte codé.



Afin de déchiffrer ces messages, les généraux ennemis se fondaient sur l'analyse fréquentielle des lettres les plus utilisées. Si le texte était chiffré en français, on pouvait deviner la lettre correspondant au « E » puisqu'elle est la plus utilisée dans la langue française. On en déduit alors le décalage. Il fallait cependant savoir en quelle langue était écrit le texte initial.



Algorithme de la fonction « codage » :

fonction codage (**phraseClair** en chaîne de caractères, **cle** en entier) retour d'une chaîne de caractères

Mettre phraseClair en majuscules

Phrase codée ← ' '

Pour chaque lettre dans phraseClair faire

Si caractère c est compris dans l'intervalle [A..Z]

ascii ← transformer caractère c en code ascii

#Méthode « **ord** »

lettre ← ascii – 65

lettre ← (lettre + cle) modulo 26

ascii ← lettre + 65

nouveau caractère ← transformer le code ascii en caractère

#Méthode « **chr** »

ajouter nouveau caractère à la phrase codée

sinon

ajouter le caractère c à la phrase codée

retourner la phrase codée

Remarque : l'algorithme de décodage est identique, la clé est simplement **négative**.

Exemple de rendu affiché dans la console : (codage du mot BONJOUR avec une clé de 3)

```

-----
1 : Codage d'une phrase
2 : Décodage d'une phrase
3 : Changement de la clé
q : quitter
-----
votre choix ?1
votre phrase en clair ?bonjour
la phrase codée est ERQMRXU
-----
1 : Codage d'une phrase
2 : Décodage d'une phrase
3 : Changement de la clé
q : quitter
-----
votre choix ?

```

| CODAGE | | | | | | | |
|--------------|---|----|----|----|----|----|----|
| Message | B | O | N | J | O | U | R |
| | 2 | 15 | 14 | 10 | 15 | 21 | 18 |
| Chiffres + 3 | 5 | 18 | 17 | 13 | 18 | 24 | 21 |
| Message codé | E | R | Q | M | R | X | U |

| DECODAGE | | | | | | | |
|----------------|---|----|----|----|----|----|----|
| Message codé | E | R | Q | M | R | X | U |
| | 5 | 18 | 17 | 13 | 18 | 24 | 21 |
| Chiffres -3 | 2 | 15 | 14 | 10 | 15 | 21 | 18 |
| Message décodé | B | O | N | J | O | U | R |

Écrire la fonction « codage » en python, puis compléter le menu qui permet de choisir l'opération à effectuer.

```

def codage(phraseClair,cle):
    phraseCodee=''
    #A compléter

cle=3
menu=''
while menu!='q':
    print("-----")
    print("1 : Codage d'une phrase")
    print("2 : Décodage d'une phrase")
    print("3 : Changement de la clé")
    print("q : quitter")
    print("-----")
    menu=input("votre choix ?")
    #A compléter

```

Exercice 6 : Chiffre de Vigenère (facultatif, pour les programmeurs aguerris)

En utilisant le code ci-dessous, programmer le codeur et décodeur du [Chiffre de Vigenère](#), amélioration du code de césar.

```

from string import ascii_uppercase as ascUp

vigenereTable = [ascUp[i:]+ascUp[:i] for i in range(len(ascUp))]

for ligne in range(0,len(vigenereTable)):
    print (ligne, vigenereTable[ligne])

```