

1 TP pour séance du 24 octobre:

1.1 Première série

1. La variable cible contient un mot de votre choix. Ecrire une boucle qui affiche une à une toutes les lettres du mot en remplaçant les *e* minuscule par un *3* (les autres lettres sont inchangées).
2. Ecrire une boucle qui remplace toutes les voyelles par la lettre *x*
3. Ecrire une boucle qui remplace une lettre sur deux par le symbole '_' (underscore)
4. Ecrire un programme qui demande à l'utilisateur un mot, et fait la transformation précédente.

```
[13]: # Question 1
cible = "permenganate"
i = 0
while (i < len(cible)):
    if cible[i] == 'e':
        print("3", end='')
    else:
        print(cible[i], end='')
    i += 1
```

p3rm3nganat3

```
[14]: # Question 2
cible = "permenganate"
i = 0
while (i < len(cible)):
    l = cible[i]
    if l == 'e' or l == 'a' or l == 'i' or l == 'o' or l == 'u':
        print("x", end='')
    else:
        print(l, end='')
    i += 1
```

pxrmxngxnxtx

à la place du test un peu laborieux, on peut écrire `l in "eaiou"`.

```
[15]: # Question 3
cible = "permenganate"
i = 0
while (i < len(cible)):
    l = cible[i]
    if i % 2 != 0:
        print("_", end='')
    else:
        print(l, end='')
    i += 1
```

p_r_e_g_n_t_

```
[16]: # Question 4
cible = input("donnez un mot")
i = 0
while (i < len(cible)):
    l = cible[i]
    if i % 2 != 0:
        print("_", end='')
    else:
        print(l, end='')
    i += 1
```

donnez un mot anticonstitutionnellement

a_t_c_n_t_t_t_o_n_l_e_e_t

1.2 Deuxième série

1. On suppose qu'on a une liste de "correspondances": chaque correspondance est une liste de deux lettres, par exemple ["e", "3"]. Demander à l'utilisateur une lettre, et indiquer (print) si cette lettre à un correspondant dans la liste.
2. Etant donnée une correspondance telle que définie à la question précédente, demander un mot à l'utilisateur, et afficher ce mot lettre par lettre en appliquant les remplacements indiqués dans la liste de correspondance.
3. A partir de deux mots de même longueur, construire une correspondance telle que définie à la question 1. Quelles précautions doit-on prendre ?
4. [Non corrigée] Demander deux mots à l'utilisateur, construire une correspondance à partir de ces deux mots; demander ensuite un nouveau mot, qui sera affiché en appliquant la correspondance.
5. [Non corrigée] Faire en sorte que l'utilisateur puisse répéter l'opération jusqu'à ce qu'il saisisse le mot fin.

```
[20]: # Question 1
correspondances = [["e", "3"], ["a", "_"], ["c", "x"]]
lettre = input("Donnez une lettre")
i = 0
while (i < len(correspondances)):
    if correspondances[i][0] == lettre:
        print(lettre, "correspond à", correspondances[i][1])
    i += 1
```

Donnez une lettre a

a correspond à _

```
[21]: # Question 2
correspondances = [["e","3"], ["a","_"], ["c","x"]]
mot = input("Donnez un mot")
i = 0
while (i < len(mot)):
    l = mot[i]
    j = 0
    while j < len(correspondances):
        if correspondances[j][0] == l:
            l = correspondances[j][1]
        j += 1
    print(l, end='')
    i += 1
```

Donnez un mot permanganate

p3rm3ng_n_t3

```
[23]: # Question 3
m1 = input("Donnez le premier mot")
m2 = input("Donnez le second mot")
if len(m1) != len(m2):
    print("Les deux mots n'ont pas la même longueur")
else:
    corr = []
    for i in range(len(m1)):
        corr.append([m1[i], m2[i]])
    print(corr)
```

Donnez le premier mot maison

Donnez le second mot bateau

[['m', 'b'], ['a', 'a'], ['i', 't'], ['s', 'e'], ['o', 'a'], ['n', 'u']]