

TP 4- Proposition de solutions

Solution 1 1. Instruction donnant le nombre de mots :

```
texte.count('')+texte.count("")+1
```

Exemple :

```
>>> texte='Pour ce qui est de l\avenir, il ne
s\agit pas de le prévoir, mais de le
rendre possible. (Antoine de Saint-Exupery)'
>>> texte.count('')+texte.count("")+1
23
```

2. Alphabet modifié :

```
>>> abc='abcdefghijklmnopqrstuvwxy'
>>> ABC=''.join([(2-i%2)*abc[i] for i in
range(len(abc))])
'aabccdeefgghiijjklmnoopqrsstuvvwxyyz'
```

Une version plus développée serait :

```
ABC=''
for i in range(len(abc)):
    if i%2==0: ABC=ABC+2*abc[i]
    else: ABC=ABC+abc[i]
```

3. Fonction donnant le nombre de voyelles :

```
nb_aeiouy=lambda mots:len([e for e in mots
if e in 'aeiouy'])
```

Exemple :

```
>>> texte='faites que le reve devore votre vie
afin que la vie ne devore pas votre reve.
(antoine de saint-exupery)'
>>> nb_aeiouy(texte)
42
```

Solution 2 Palindrome

1. Fonction d'identification d'un palindrome :

```
def palindrome(texte):
    '''
    Entrer : texte est une chaine de caracteres
    Sortie : boleen testant si la chaine est un
    palindrome
    '''
    texte=lettre(texte)
    for i in range(len(texte)//2):
        if texte[i]!=texte[-i-1]:
            return False
    return True
```

Exemple :

```
>>> palindrome('lol')
True
>>> palindrome('Nurses run')
Remplacer N par : n
Remplacer par :
True
```

2. Fonction interactive normalisant le texte en minuscule utilisant un dictionnaire :

```
def lettre(mots):
    '''
    Entree : mots est une chaine de caracteres
    Sortie : une chaine de caracteres qui ne contient q
    '''
    D,c={},''
    for e in mots:
        if e in D:
            c=c+D[e]
        else:
            r=input('Remplacer '+e+' par : ')
            c=c+r
            D[e]=r
    return c
```

Solution 3 Recherche d'un mot

Recherche d'un mot

```
def recherche(mot, texte):
    '''
    Entree : deux chaines de caracteres
    Sortie : liste des positions ou la premiere
    est inseree dans la deuxieme
    '''
    L=[]
    for i in range(0, len(texte)-len(mot)+1):
        j=0
        while j<len(mot) and texte[i+j]==mot[j]:
            j=j+1
        if j==len(mot):
            L.append(i)
    return L
```

Exemple :

```
>>> recherche('aa', 'aaaabaa')
[0, 1, 2, 5]
```