

## Exercice 1 – Ordre lexicographique

<http://mathematice.fr/> – Serge Bays

L'objectif est d'écrire un programme qui trie une liste de mots et les range dans l'ordre lexicographique (ordre des dictionnaires).

1. Ecrire la définition de la variable "alphabet": `alphabet="AaàBbCcDdEeéeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuùVvWwXxYyZz"`
2. Ecrire une fonction `ordre_alphabetique(c1, c2)` qui prend en argument deux caractères alphabétiques et renvoie -1 si `c1` est avant `c2`, 1 si `c2` est avant `c1` et 0 si `c1 = c2`. On pourra utiliser la méthode `index` qui renvoie l'indice d'un élément dans une chaîne de caractère.
3. Ecrire une fonction `ordre_lexicographique(m1, m2)` qui prend en argument deux mots et renvoie -1 si "`m1` < `m2`" pour l'ordre lexicographique, 0 si "`m1` = `m2`" et 1 si "`m1` > `m2`". On utilisera la fonction `ordre_alphabetique`.
4. Ecrire une fonction `tri_lexicographique(liste)` qui prend en argument une liste de mots et renvoie la liste triée. On utilisera la fonction `ordre_alphabetique` et l'algorithme du tri par insertion.

**Correction** `alphabet='AaàBbCcDdEeéeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuùVvWwXxYyZz'`

```
def ordre_alphabetique(c1,c2):
    if alphabet.index(c1)<alphabet.index(c2):
        return -1
    elif alphabet.index(c2)<alphabet.index(c1):
        return 1
    else:
        return 0

def ordre_lexicographique(m1,m2):
    n=min(len(m1),len(m2))
    for i in range(n):
        if ordre_alphabetique(m1[i],m2[i])==-1:
            return -1
        elif ordre_alphabetique(m1[i],m2[i])==1:
            return 1
    if len(m1)<len(m2):
        return -1
    elif len(m2)<len(m1):
        return 1
    else:
        return 0

def tri_lexicographique(liste):
    for i in range(len(liste)-1):
        k = i+1 # indice de la cle
        cle=liste[k]
        while ordre_lexicographique(cle,liste[k-1])==-1 and k>0:
            liste[k] = liste[k-1]
            k = k-1
        liste[k]=cle
        print("etape",i," :\t",liste)
    return liste

t=['moi', 'toi', 'bonjour', 'salut', 'bon']
print("liste :\t\t",t)
print("resultat : \t",tri_lexicographique(t))
```