

09. Recherche d'un mot dans une chaîne de caractères

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.

Exercice 1. Tester le programme suivant :

```
def extraire(l,a,b):
    '''extraction des caractères d'indices compris entre a et b
    dans la chaîne de caractères l'''
    m=""
    for k in range(a,b+1):
        m=m+l[k]
    return(m)
```

Exercice 2. Créer puis tester une fonction *renverser* de la variable *l* qui retourne la chaîne de caractères obtenue en lisant les caractères de la chaîne *l* dans l'ordre décroissant de leurs indices.

(par exemple *renverser("python")* vaut *"nohtyp"*)

Exercice 3. Créer puis tester une fonction *debut* des variables *l* et *m* permettant de tester si la chaîne de caractères *l* débute par le mot *m*, on pourra utiliser la fonction *extraire*.

(par exemple *debut("python", "py")* vaut *True*)

Exercice 4. Créer puis tester une fonction *fin* des variables *l* et *m* permettant de tester si la chaîne de caractères *l* finit par le mot *m*.

(par exemple *fin("python", "thon")* vaut *True*)

Exercice 5. Créer puis tester une fonction *occurrences* des variables *l* et *m* retournant le nombre d'occurrences du mot *m* dans la chaîne de caractères *l*.

(par exemple *occurrences("montypython", "on")* vaut 2)

Exercice 6. Créer puis tester une fonction *atteindre* des variables *l* et *c* retournant la liste des indices des occurrences du caractère *c* dans la chaîne de caractères *l*.

(par exemple *atteindre("montypython", "y")* vaut [4, 6])

Exercice 7. Créer puis tester une fonction *remplacer* des variables *l*, *c1* et *c2* retournant la chaîne de caractères obtenue en remplaçant les occurrences du caractère *c1* par le caractère *c2* dans la chaîne de caractères *l*.

(par exemple *remplacer("montypython", "y", "i")* vaut *"montipithon"*)

Exercice 8. Créer puis tester une fonction *statistiques* de la variable *l* retournant la liste des caractères de la chaîne *l* ainsi que la liste de leurs nombres d'occurrences respectifs.

(par exemple *statistiques("montypython")* vaut ['m','o','n','t','y','p','h'], [1, 2, 2, 2, 2, 1, 1])

Réponses

- 2) `def renverser(l):`
 '''retourne la chaîne de caractères obtenue en lisant les caractères de la chaîne l dans l'ordre décroissant de leurs indices'''
 L=""
 for k in range(0,len(l)):
 L=L+l[len(l)-1-k]
 return(L)
- 3) `def debut(l,m):`
 '''teste si la chaîne de caractère l débute par le mot m'''
 if extraire(l,0,len(m)-1)==m:
 return(True)
 else:
 return(False)
- 4) `def fin(l,m):`
 '''teste si la chaîne de caractère l finit par le mot m'''
 if extraire(l,len(l)-len(m),len(l)-1)==m:
 return(True)
 else:
 return(False)
- 5) `def occurrences(l,m):`
 '''calcul du nombre d'occurrences du mot m dans la chaîne de caractères l'''
 s=0
 for k in range(0,len(l)-len(m)+1):
 if extraire(l,k,k+len(m)-1)==m:
 s=s+1
 return(s)
- 6) `def atteindre(l,c):`
 '''retourne la liste des indices des occurrences du caractère c dans la chaîne de caractères l'''
 L=[]
 for k in range(0,len(l)):
 if l[k]==c:
 L.append(k)
 return(L)
- 7) `def remplacer(l,c1,c2):`
 '''retourne la chaîne de caractères obtenue en remplaçant les occurrences du caractère c1 par le caractère c2 dans la chaîne de caractères l'''
 L=""
 for k in range(0,len(l)):
 if l[k]==c1:
 L=L+c2
 else:
 L=L+l[k]
 return(L)
- 8) `def statistiques(l):`
 '''retourne la liste des caractères de la chaîne l ainsi que la liste de leurs nombres d'occurrences respectifs'''
 L=l #chaîne en cours de traitement
 L1=[] #liste des caractères
 L2=[] #liste des nombres d'occurrences
 while(len(L)>0):
 c=L[0] #caractère en cours de traitement
 s=1
 Ltemp=""
 for k in range(1,len(L)):
 if L[k]==c:
 s=s+1
 else:
 Ltemp=Ltemp+L[k]
 L1.append(c)
 L2.append(s)
 L=Ltemp
 return(L1,L2)