

Exercice 1 (len, min, max)

Ecrire une fonction qui retourne la longueur d'une chaîne de caractères passée en paramètre. (slen)

Exemple : slen(«Hello») -> 5

```
def slen(s):
    l=0
    for c in s:
        l+=1
    return l
```

Ecrire une fonction qui retourne le caractère minimal d'une chaîne de caractères passée en paramètre. (smin)

Exemple : smin(«Hello») -> «o»

```
def smin(s):
    assert slen(s)>0, "chaîne vide"
    m=s[0]
    for c in s:
        if c<m:
            m=c
    return m
```

Ecrire une fonction qui retourne le caractère maximal d'une chaîne de caractères passée en paramètre. (smax)

Exemple : smax(«Hello») -> «o»

```
def smax(s):
    assert slen(s)>0, "chaîne vide"
    m=s[0]
    for c in s:
        if c>m:
            m=c
    return m
```

Exercice 2 (Chaîne carrée)

« bonbon » est chaîne carrée. Ecrire une fonction qui prend en paramètre une chaîne de caractères et retourne True si la chaîne est carrée False sinon.

Exemple : carre(«bonbon») -> True

```
def carre(s):
    return s[:len(s)//2]==s[len(s)//2:]
```

Exercice 3 (Inverse d'une chaîne)

Ecrire une fonction inverse qui prend en paramètre une chaîne de caractères et retourne une nouvelle chaîne de caractères qui soit son inverse.

Exemple : inverse(«Top») -> «poT»

```
def inverse(s):
    r=""
    for c in s:
        r=c+r
    return r
```

Exercice 4 (Palindrome)

Une chaîne de caractères est dite palindrome s'elle se lit de la même façon des deux côtés (s'elle est symétrique par rapport à son centre). Ecrire une fonction qui prend en paramètre une chaîne de caractères et retourne True si la chaîne est palindrome False sinon.

Exemple : palindrome(«radar») -> True

```
1ère méthode : en utilisant la fonction inverse
def palindrome(s):
    return s==inverse(s)
```

2^{ème} méthode en utilisant une boucle

```
def palindrome(s):
    for i in range(len(s)//2):
        if s[i]!=s[-i-1]:
            return False
    return True
```

Exercice 5 (Words Counter)

Ecrire une fonction wcount qui prend en paramètre une chaîne de caractères et retourne le nombre de mots de cette chaîne.

Exemple : wcount(« beauty is a mind thinking way ») -> 6

```
def words_count(s):
    if s[0]==" ":
        c=0
    else:
        c=1
    for i in range(len(s)-1):
        if s[i]==" " and s[i+1]!=" ":
            c=c+1
    return c
```

Exercice 6 (Tri de chaînes)

Ecrire une fonction stri qui prend en paramètre une chaîne de caractères et retourne une nouvelle chaîne de caractères MAJUSCULE triée par ordre alphabétique.

Exemple : stri(«Top») -> «OPT»

```
def stri(s):
    s=s.upper()
    r=""
    while s!="":
        i=imin(s)
        r=r+s[i]
        s=s[:i]+s[i+1:]
    return r

def imin(s):
    im=0
    m=s[0]
    for i in range(len(s)):
        if s[i]<m:
            m=s[i]
            im=i
    return im
```

Exercice 7 (Tri de sous chaînes)

Ecrire une fonction sbtri qui prend en paramètre une chaîne de caractères MAJUSCULE S et un entier naturel L retourne la plus petite chaîne de caractères parmi toutes chaînes de caractères formées par seulement le tri de L caractères de la chaîne d'origine.

Exemple : S = "ESPRIT" i=0, stri(«ESP») -> «EPS», «EPSRIT» i=1, stri(«SPR») -> «PRS», «EPRSIT» i=2, stri(«PRI») -> «IRP», «ESIRPT» i=3, stri(«RIT») -> «IRT», «ESPIRT»

Donc, sbtri(« ESPRIT », 3) -> «EPRSIT»

Vous allez utiliser la fonction « **stri** » de l'exercice précédent.