Chaines de caractères

Mise en route

1. Tester le code suivant et comprendre à quoi correspond l'affichage obtenu :

```
s = "Bonjour"
print(len(s))
print(s[0])
print(s[6])
```

2. Modifier le programme en remplaçant la chaine **s** du début par

```
s = "Bye"
```

et noter ce qu'il se passe lors de l'exécution du code.

3. Modifier le code pour qu'il affiche le tout premier élément apparaissant dans la chaine **s** ainsi que le tout dernier (votre code doit bien entendu fonctionner pour toute chaine **s** écrite au début, à partir du moment où elle contient au moins un élément).

Opérations sur les chaines

4. Tester les codes suivants :

```
s1 = "Bonjour"
s2 = "Bye"
s = s1+s2
print(s)
```

```
s1 = 3*"a"
s2 = 2*"Bonjour"
print(s1)
print(s2)
```

et en déduire à quoi correspondent les opérations **+** et ***** appliquées aux chaines.

5. Quel est le code permettant de construire la chaine `"abab...ab"` où chaque lettre a et b apparait 20 fois ?

Parcours d'une chaine

6. Tester le code suivant et comprendre à quoi correspond l'affichage obtenu :

```
s = "abcdeabcde"
t = ""
for k in range(0,len(s)):
    t = t+s[k]
print(t)
```

7. Modifier le code pour que la construction de la chaine `t` ne prenne en compte que les caractères autres que `e` et vérifier que le résultat affiché est conforme.

8. Modifier ce code afin d'écrire une fonction de la forme :

```
def supprimer_espaces(s):
    ...
    return t
```

qui prend en paramètre une chaine `s` et qui construit et renvoie la chaine `t` obtenue à partir de `s` en ne conservant que les caractères autres que `" "` (caractère espace). Tester avec :

```
print(supprimer_espaces("Bonjour tout le monde !"))
```

et vérifier que le résultat affiché est conforme.

9. À partir du code donné en question 6, écrire une fonction

```
def doubler(s):
    ...
    return t
```

qui construit et renvoie la chaine `t` contenant les mêmes caractères que `s` mais chacun apparaissant 2 fois. Vérifier que `print(retourner("Hello"))` affiche `HHeellllloo`.

10. Toujours à partir du code de la question 6, écrire une fonction

```
def retourner(s):
    ...
    return t
```

qui construit et renvoie la chaine `t` contenant les mêmes caractères que `s` mais dans l'ordre inverse. Vérifier que `print(retourner("Hello"))` affiche `olleH`.

Transformer des chaines en nombres et réciproquement

11. Tester les codes suivants :

```
s = str(125)
print(s)
s = str([1,2,3,4])
print(s)
```

```
s = "125"
x = int(s)
print(x)
print(x+5)
```

Ceci peut parfois servir dans les programmes.

Recherche d'un élément

12. Tester le code suivant et comprendre à quoi correspond l'affichage obtenu :

```
s = "abcdeabcde"
k = 0
while k<len(s) and s[k]!="d":
    k = k+1
print(k)
```

Quelle est la valeur k affichée ? Quelle est la valeur $s[k]$ correspondante ?

Réponse : _____

13. Recommencer en prenant au départ la chaine `s="abceabce"`. Quelle est la valeur k affichée ?

Réponse : _____

14. Écrire une fonction `positions(c,s)` qui construit et renvoie la liste `L` constituée des positions où le caractère `c` apparait dans la chaine `s`. Vérifier par exemple que :

```
print(positions("a","abcabc"))
print(positions("d","abcabc"))
```

affichent respectivement la liste `[0,3]` et la liste `[]`.

15. Écrire une fonction `recherche(c,s)` qui renvoie `True` si le caractère `c` apparait dans la chaine `s` et renvoie `False` dans le cas contraire. Tester avec le code :

```
print(recherche("a","abcabc"))
print(recherche("d","abcabc"))
```

Réponse : _____

Corrections

Q 3.

```
s = "Hello"
print(s[0])
print(s[len(s)-1])
```

```
H o
```

Q 5.

```
print(20*"ab")
```

```
abababababababababababababababababababab
```

Q 7.

```
s = "abcdeabcde"
t = ""
for k in range(0,len(s)):
    if s[k]!="e":
        t = t+s[k]
print(t)
```

```
abcdabcd
```

Q 8.

```
def supprimer_espaces(s):
    t = ""
    for k in range(0,len(s)):
        if s[k]!=" ":
            t = t+s[k]
    return t

print(supprimer_espaces("Bonjour tout le monde !"))
```

```
Bonjourtoutlemonde!
```

Q 9.

```
def doubler(s):
    t = ""
    for k in range(0,len(s)):
        t = t+2*s[k]
    return t

print(doubler("Hello"))
```

```
HHeellllloo
```

Q 10.

```
def retourner(s):
    t = ""
    for k in range(0,len(s)):
        t = s[k]+t
    return t

print(retourner("Hello"))
```

```
olleH
```

Q 14.

```
def positions(c,s):
    L = []
    for k in range(0,len(s)):
        if s[k]==c:
            L.append(k)
    return L

print(positions("a","abcabc"))
print(positions("d","abcabc"))
```

```
[0, 3] []
```

Q 15.

```
def recherche(c,s):
    L = []
    for k in range(0,len(s)):
        if s[k]==c:
            return True
    # Boucle terminée
    return False

print(recherche("a","abcabc"))
print(recherche("d","abcabc"))
```

```
True False
```