

Fiche de TD numéro 4 - Les chaînes de caractères

Exercice 1 : Spécifiez, puis écrivez une fonction `nbre_occ` qui retourne le nombre d'occurrences d'un caractère dans une chaîne de caractères.

```
>>> nbre_occ("bonjour", "z")
0
>>> nbre_occ("abracadabra", "a")
5
```

Exercice 2 : Spécifiez puis écrivez la fonction `extrait` qui prend en paramètre une chaîne de caractères `ch`, un indice de début `deb` et un nombre de caractères `n`. Cette fonction retourne la chaîne de caractères constituée de `n` caractères de `ch` à partir de l'indice `deb`.

```
>>> extrait("bonjour", 2, 4)
'njou'
>>> extrait("bonjour", 2, 18)
'njour'
>>> extrait("bonjour", 16, 2)
''
>>> extrait("bonjour", -1, 2)
''
```

Exercice 3 : Spécifiez puis écrivez la fonction `premiers` qui retourne la sous-chaîne d'une chaîne contenant ses `n` premiers caractères.

```
>>> premiers('bonjour', 2)
'bo'
```

Exercice 4 : Spécifiez puis écrivez la fonction `derniers` qui retourne la sous-chaîne d'une chaîne contenant ses `n` derniers caractères.

```
>>> derniers('bonjour', 3)
'our'
```

Exercice 5 : Spécifiez, puis écrivez une fonction `est_valide` qui prend en paramètres deux chaînes de caractères, un *mot* et un *schéma*. Le schéma comporte des caractères `'?'` qui correspondent à des jokers. Un mot est valide par rapport à un schéma, s'il est identique au schéma sur tous les caractères, sauf sur les `'?'` où n'importe quel caractère peut être accepté.

```
>>> estValide("mémé", "m?m?")
True
>>> estValide("momi", "m?m?")
True
>>> estValide("mmmmh", "m?m?")
False
```

Exercice 6 : Spécifiez puis écrivez une fonction `est_prefixe` qui prend en paramètres deux chaînes de caractères et dit si la première est un préfixe de l'autre.

```
>>> est_prefixe("pre", "prefixe")
True
>>> est_prefixe("pro", "prefixe")
False
>>> est_prefixe("fi", "prefixe")
False
```

Exercice 7 : Spécifiez puis écrivez une fonction `est_suffixe` qui prend en paramètres deux chaînes de caractères et dit si la première est un suffixe de l'autre.

```
>>> est_suffixe("fixe", "suffixe")
True
>>> est_suffixe("fix", "suffixe")
False
>>> est_suffixe("suffixe", "suffixe")
True
```

Exercice 8 : Spécifiez puis écrivez une fonction `contient_dans_ordre` qui prend en paramètres deux chaînes de caractères et dit si toutes les lettres de la première sont présentes dans l'autre dans le même ordre.

```
>>> contient_dans_ordre("aeiou", "blableblibloblublublu")
True
>>> contient_dans_ordre("aeiyu", "blableblibloblublublu")
False
```

Exercice 9 : Spécifiez puis écrivez une fonction `miroir` qui retourne la chaîne miroir d'une chaîne donnée.

```
>>> miroir("machin")
'nihcma'
```

Exercice 10 : Spécifiez et écrivez une fonction `palindrome` qui teste une chaîne de caractères donnée est ou non un palindrome.

```
>>> palindrome("toto")
false
>>> palindrome("laval")
true
```