

# Ctrl+F, Ctrl+R

## I Rechercher encore

Il existe un grand nombre d'algorithmes ayant pour but rechercher de chaîne de caractères dans un texte. On se propose d'utiliser celui de la fenêtre glissante.

Pour chercher un mot de taille  $n$  dans un texte de taille  $m$ , on vérifie pour chaque position  $i < m-n$  du texte si les caractères  $i, i+1, \dots, i+n-1$  forment le mot recherché.

### Exercice 1. Rechercher à une position

Implémentez une fonction `rechercher_i` qui prend trois entrées : une chaîne de caractère  $s$ , une chaîne de caractère  $w$  et un entier  $i$  et qui renvoie un booléen vrai si et seulement si le mot  $w$  est une sous-chaîne de  $s$  à l'indice  $i$  de  $s$ . On supposera que  $0 < \text{len}(w) \leq \text{len}(s)-i$ .

Testez votre fonction pour les entrées suivantes :

1.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dine"$ ,  $i = 2$
2.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dine"$ ,  $i = 20$
3.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dine"$ ,  $i = 1$
4.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dina"$ ,  $i = 2$
5.  $s = "Nadine"$ ,  $w = "Nadine"$ ,  $i = 0$
6.  $s = "Yooooohoooo\|!"$ ,  $w = "ooo"$ ,  $i = 1$
7.  $s = "Yooooohoooo\|!"$ ,  $w = "ooo"$ ,  $i = 2$
8.  $s = "Yooooohoooo\|!"$ ,  $w = "ooo"$ ,  $i = 8$

### Exercice 2. Rechercher à l'aide de recherche\_i

Implémentez une fonction `rechercher_0` qui prend trois entrées : une chaîne de caractère  $s$ , une chaîne de caractère  $w$  et un entier  $d$  et qui cherche le mot  $w$  dans le texte  $s$  à partir du caractère  $d$ 'indice  $d$ . La sortie de la fonction sera l'indice le plus petit de  $s$  où se trouve la première lettre du mot  $w$  dans  $s$ ,  $-1$  si  $w$  n'est pas dans  $s$ .

Pour implémentez cette fonction vous utiliserez la fonction `rechercher_i` de l'exercice précédent.

Testez votre fonction pour les entrées suivantes :

1.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dine"$ ,  $d = 0$
2.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dine"$ ,  $d = 20$
3.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dine"$ ,  $d = 2$
4.  $s = "Nadine\|aime\|la\|grenadine"$ ,  $w = "dina"$ ,  $i = 1$
5.  $s = "Nadine"$ ,  $w = "Nadine"$ ,  $d = 0$
6.  $s = "Yooooohoooo\|!"$ ,  $w = "ooo"$ ,  $d = 0$
7.  $s = "Yooooohoooo\|!"$ ,  $w = "ooo"$ ,  $d = 2$
8.  $s = "Yooooohoooo\|!"$ ,  $w = "ooo"$ ,  $d = 8$

### Exercice 3. Rechercher avec boucle imbriquée

Implémentez une fonction `rechercher` qui prend trois entrées : une chaîne de caractère  $s$ , une chaîne de caractère  $w$  et un entier  $d$  et qui cherche le mot  $w$  dans le texte  $s$  à partir du caractère  $d$ 'indice  $d$ . La

sortie de la fonction sera l'indice le plus petit de s où se trouve la première lettre du mot w dans s, -1 si w n'est pas dans s.

Testez votre fonction pour les entrées données dans la question précédentes.

#### **Exercice 4. Rechercher tout**

En utilisant la fonction `rechercher` de la question précédente, implémentez une fonction `rechercher_tout` qui prend deux entrées : une chaîne de caractère s et une chaîne de caractère w. La sortie de la fonction sera la liste des indices de s où se trouve la première lettre d'un mot w dans s.

*On fera attention que deux mots w ne se superpose pas.* Si c'est le cas on prendra uniquement l'indice le plus petit. e.g pour le texte : "nananadine", si on cherche "nana" on trouvera deux positions, 0 et 2. Mais la lettre en position 2 est déjà une lettre de la première occurrence de « nana », on renverra uniquement la position 0. Cela évitera les conflits lorsqu'il s'agira de remplacer le mot "nana"

Testez votre fonction pour les entrées suivantes :

1. s = "Nadine\u00e0ime\u00e0la\u00e0grenadine", w = "dine"
2. s = "Nadine\u00e0ime\u00e0la\u00e0grenadine", w = "dina"
3. s = "Nadine", w = "Nadine"
4. s = "Yooooohoooo\u00e0!", w = "ooo"

#### **Exercice 5. Remplacer**

En utilisant la fonction `rechercher_tout` de la question précédente, implémentez une fonction `remplacer` qui prend trois entrées : une chaîne de caractère s, une chaîne de caractère w et une chaîne de caractère v. La sortie de la fonction sera le texte dans lequel toutes les occurrences de w seront remplacées par celle de v.

*Il existe plusieurs manières de résoudre ce problème.* Les indications ci-dessous ne sont pas forcément utiles pour toutes les solutions.

*On pourra commencer par supposer que w et v sont de même longueurs.* Pour ensuite gérer le cas où v et w sont de longueurs différentes on conservera le décalage induit sur les indices par les remplacements déjà effectués.

On peut passer d'une chaîne de caractère à une liste de caractère grâce à `list(s)`. On peut aussi passer d'une liste de caractère à une chaîne de caractère en faisant '''.join(L). On peut remplacer plusieurs éléments consécutifs d'une liste par une autre liste grâce aux tranches : L[12:21] = [1,2,4].

Testez votre code à l'aide de l'exemple suivant :

```
s8="Après\u00e0un\u00e0premier\u00e0jeu ,\u00e0les\u00e0enfants\u00e0jouent\u00e0un\u00e0nouveau\u00e0jeu "
w8="jeu"
remplacer(s8,w8,"joujou")
```

## **II Fichiers**

#### **Exercice 6. Lecture d'un fichier**

1. Quel est le chemin absolu du fichier dans lequel vous travaillez ?

2. Quel est le chemin absolu du fichier `star_wars.txt` disponible dans le dossier `txt` du dossier de ressources extrait sur votre ordinateur ?
3. Quel est le chemin relatif du fichier `star_wars.txt` si on se situe dans le dossier de ressources extrait sur votre ordinateur ?

Implémentez les fonctions suivantes :

1. `nb_c` prend en entrée un chemin de fichier et renvoie le nombre de caractères écrits dans le fichier.
2. `nb_m` prend en entrée un chemin de fichier et renvoie le nombre de mots écrits dans le fichier.
3. `nb_l` prend en entrée un chemin de fichier et renvoie le nombre de lignes écrites dans le fichier.

#### **Exercice 7. Recherche dans un fichier**

En utilisant la console SPYDER et la fonction `rechercher_tout` donnez la liste des positions où on peut trouver le mot "nadine" dans le fichier "txt/texte\_interessant.txt".

#### **Exercice 8. Remplacer dans un fichier**

En utilisant la console SPYDER et la fonction `remplacer` remplacer toutes les occurrences du mot "Force" par "Nadine" dans le fichier "txt/star\_wars.txt" et écrivez un fichier "txt/nadine\_wars.txt" avec le résultat.