

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
30/09/2021	2 - Algorithmes à boucles imbriquées	TD 2-1 – Facteurs dans un texte

Informatique

2

Algorithmes à boucles imbriquées

TD 2-1

Facteurs dans un texte

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
30/09/2021	2 - Algorithmes à boucles imbriquées	TD 2-1 – Facteurs dans un texte

Exercice 1: Recherche d'un mot dans un texte

La liste [1, 3, 5] peut être vue comme une sous-liste de la liste [15, -12, 1, 3, 5, 19, 23] (ici par sous-liste on entend « en un seul morceau ». Ainsi, [1, 3, 5] n'est pas vue comme une sous-liste de [1, 2, 3, 4, 5]. De même « tar » est un sous-mot de « il est tard ».

On cherche à déterminer si un motif apparaît dans une liste ou dans un texte. Les manipulations de listes et de chaînes de caractères étant proches, on demande d'écrire des codes qui vont fonctionner indifféremment sur les deux types de structure.

Voici la démarche naïve de recherche d'un motif dans un texte qui sera programmée :

- On effectue la recherche avec une boucle qui va considérer toutes les positions possibles pour le mot (on appelle position du mot l'indice python de la 1^{ère} lettre du mot).
- Pour chaque position i, on teste si le mot apparaît en position i :
 - o Dans la 1^{ère} fonction on autorise les slices (voir explication ci-dessous) pour cette comparaison
 - o Dans la 2^{ème} fonction on autorise uniquement des comparaisons lettre à lettre

NB : Le terme anglais slice est associé à l'idée de découpage. En langage Python, cela permet d'extraire une tranche d'un objet itérable. Si L est une liste ou une chaîne de caractères, L[i:j] renvoie une nouvelle liste ou chaîne de caractère composée des éléments d'indices python i à j-1 inclus de L. Par exemple :

```
>>> s='il est tard'
>>> s[3:8]
'est t'
```

Question 1: En notant T la longueur du texte et M la longueur du mot, quelles sont les valeurs de i à considérer pour tester toutes les positions possibles du mot dans le texte ? Comment récupérer à l'aide d'un slice le sous-mot du texte qui se trouve à la position i et qu'il faudra comparer avec mot ?

Question 2: Ecrire une fonction Recherche_motif_1(mot,texte) qui, étant donnés deux chaînes de caractères (ou deux listes) mot et texte, renvoie True si le mot est présent et False sinon, slices autorisés

Vérifier :

```
>>> Recherche_motif_1('ard','il est tard')
True

>>> Recherche_motif_1([1,3,5],[15,-12,1,3,5,19,23])
True

>>> Recherche_motif_1([1,3,5],[1,2,3,4,5])
False
```

Question 3: Ecrire une fonction Recherche_motif_2(mot,texte) qui, étant donnés deux chaînes de caractères (ou deux listes) mot et texte, renvoie True si le mot est présent et False sinon, slices non autorisés

Remarques :

- Seules les comparaisons lettre à lettre sont autorisées
- On utilisera deux boucles imbriquées
- Réfléchissez à la complexité de votre code et faites en sorte que tout le motif ne soit pas systématiquement testé inutilement

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
30/09/2021	2 - Algorithmes à boucles imbriquées	TD 2-1 – Facteurs dans un texte

Question 4: Ecrire une fonction `Positions_motif(mot, texte)` qui renvoie la liste, éventuellement vide, des positions du mot dans le texte, slices autorisés

Vous avez à votre disposition le fichier « 3-1 - TD - Recherche d'un élément - Extrait.py » dont [voici le lien](#).

Question 5: Récupérer (copier-coller) la chaîne de caractères extrait dans le fichier et déterminer combien de fois le mot « son » apparaît, puis le mot « envol »

Réponses : « son » 11 fois, pas d'envol