Recherche d'un mot dans une chaîne de caractères

On considère $A=a_1 \dots a_n$ et $B=b_1 \dots b_p$ deux chaines de caractères non vides.

- On appelle **mot** de A toute chaine de caractères où $a_{i_1} \dots a_{i_k}$ tels que $1 \leqslant i_1 < \dots < i_k \leqslant n$, les caractères étant **consécutifs** dans A.
- On appelle plus long mot commun à A et B tout mot commun à A et B de longueur maximale.
- Si l'une des chaines A ou B est vide, ou si A et B n'ont aucun mot commun, on convient que la chaine vide (c'est-à-dire : ' ') est l'unique plus long mot commun à A et B.

Par exemple :

- les chaines de caractères "AAA" et "TAA" sont des mots communs aux chaines de caractères chaine1 = "ATAAAGA" et chaine2 = "TAAACA".
- La chaine de caractères sschaine = "ATGC" est une plus longue sous-chaine commune aux chaines de caractères chaine1 = "AATGCG" et chaine2 ="TATTATGC".

0 Questions préliminaires

1. Quels sont les mots communs aux chaines "AATGCG" et "TATTAGC" ?
2. Déterminer les plus longs mots communs (ceux qui ont le nombre maximal de caractères) aux chaines "GATGCAAT" et "CAATGCA" ?

I Recherche d'un mot dans une chaîne de caractères

Le but est de savoir si un mot fait partie d'une chaîne de caractères

1. À l'aide du doc. 1, compléter le script ci-dessous définissant la fonction **recherche_mot** prenant comme argument deux chaînes de caractères **ch** et **mot** renvoyant **True** si la chaîne mot est contenu dans la chaîne **ch** et **False** sinon.

```
def recherche_mot(ch, mot):

    n = len(ch)
    m = len(mot)

    for ind in range(......):
        nb = 0

        while ...... and ..... :

            ........................

        if .....:

            return True

    return False
```

p	i	t	a	p	i	p	a	p	a	ind=0, nb=1
p	i	t	a	p	i	p	a	p	a	ind=0, nb=2
p	i	t	a	p	i	p	a	p	a	ind=0, nb=2
p	**i**	t	a	p	i	p	a	p	a	ind=1, nb=0
p	i	**t**	a	p	i	p	a	p	a	ind=2, nb=0
p	i	t	**a**	p	i	p	a	p	a	ind=3, nb=0
p	i	t	a	**p**	i	p	a	p	a	ind=4, nb=1
p	i	t	a	**p**	i	p	a	p	a	ind=4, nb=2
p	i	t	a	**p**	i	p	a	p	a	ind=4, nb=3
p	i	t	a	**p**	i	p	a	p	a	ind=4, nb=4

Doc 1 *L'illustration permet de visualiser la recherche du mot 'pipa' dans la chaîne 'pitapipapa'. Les cases rose indiquant un arrêt.*

2. En utilisant la fonction **recherche_mot**, proposer un script définissant la fonction **mot_commun** prenant comme arguments trois chaînes de caractères **ch1**, **ch2** et **mot** renvoyant **True** si la chaîne **mot** est commune aux chaînes **ch1** et **ch2** et **False** sinon.

II Mots communs à deux chaînes

Le but de cette question est de dresser la liste de tous les mots communs à deux chaînes de caractères.

1. Le but de cette question est de dresser la liste de tous les mots communs à deux chaînes de caractères.

   a. Compléter le script suivant afin que la fonction **Liste_Mot**s renvoie la liste de tous les mots que l'on peut composer à partir de la chaîne **ch**.

```
def Liste_Mots(ch):
    L=['']
    l=len(ch)
    for i in range(...,....):
        for k in range(.......):
            L+=[ch[k:i+k]]
    return L
```

   b. En utilisant la fonction **Liste_Mots**, proposer un script définissant la fonction **Liste_Mots_Communs** prenant comme argument deux chaînes de caractères **ch1** et **ch2** et renvoyant la liste de mots communs aux chaînes **ch1** et **ch2**.

   On pourra utiliser la fonction **supp_tri** du TP6 pour supprimer les éventuels mots en doublons.

2. En utilisant la fonction **Liste_Mots_Communs** proposer un script définissant la fonction **Liste_Mots_Communs_lgmax** prenant comme argument deux chaînes de caractères **ch1** et **ch2** et renvoyant la liste des mots de longueur maximale communs aux chaînes **ch1** et **ch2.**

3. APPLICATION

a. Proposer une fonction **chaine_ADN** qui admet pour argument un entier *n* et qui renvoie une chaîne ADN une chaîne de caractères composée de 'G', 'A', 'T' et 'C' choisis de manière aléatoire.
On pourra utiliser la fonction **randint(n,m)** de la bibliothèque **random** qui renvoie de manière un entier aléatoire entre n et m.

b. Proposer une fonction **chaines_ADN_texte** qui admet pour arguments deux entiers *n* et *m* et qui renvoie un fichier chaines_ADN.txt composé de *m* lignes représentant chacune une chaîne ADN aléatoire de longueur *n* .

c. i) Proposer une fonction **chaines_ADN_communes_max** qui lit chaque ligne d'un fichier chaines_ADN.txt **ne comportant que 2 lignes**, extrait les chaines ADN sous forme de chaines de caractères et renvoie la liste des plus longues sous-chaînes communes aux deux chaînes sous la forme d'un fichier chaines_communes_max.txt avec une chaîne commune par ligne.

ii) **Plus difficile** Modifier la fonction précédente pour un fichier chaines_ADN.txt **comportant m lignes.**