TP4 : Recherche de motifs dans l'ADN

L'objectif de ce TP est d'impanter l'algorithme de recherche de motif dans un texte vu en cours, pour recherche un motif dans une séquence ADN. Le programme doit afficher (au choix de l'utilisateur) "Oui/Non" pour dire si le motif apparaît, le nombre de fois où le motif apparaît, ou la liste des indices auquel le motif apparaît. On accepte les faux positifs, et on ne cherche pas à les *corriger*.

Le programme rendu (avec son code source bien évidemment) doit prendre en entrée deux arguments `TEXTE` et `MOTIF` : ces chaînes de caractères sont les noms des fichiers qui contiennent respectivement le texte et le motif à rechercher. Ainsi, `./mon_super_programme texte.txt motif.txt` doit renvoyer "Oui" (par exemple) si le motif contenu dans le fichier `motif.txt` apparaît dans `texte.txt`.

Le format des fichiers d'entrée est une suite de lettres ("A", "C", "G" ou "T") séparées éventuellement par des espaces ou retour à la ligne (qui doivent être ignorés).

Quelques indications

- Pour traiter des grands fichiers, on aura besoin de grands entiers. En C/C++, il est conseillé de travailler avec la bibliothèque GMP qui permet cela.
- L'algorithme travaille avec des nombres premiers. Pour tirer aléatoirement un nombre premier, deux solutions vous sont proposées : soit vous utilisez GMP et lisez la doc, soit je fournis des fichiers `nombres_premiers_i` qui contiennent chacun une liste de 100000 nombres premiers inférieurs à 2^i.
- Essayez dans un premier temps de coder l'algorithme avec des entiers *standards* (`int`, `long`, etc.) et de l'appliquer à des petits exemples. Faites une version pour les grands entiers après coup.

Fichiers de test

En plus des fichiers `nombres_premiers_i`, je fournis une séquence d'ADN (séquençage du chromosome 18 chez l'humain[1]), ainsi que des sous-séquences de cette séquence. Un fichier *plus petit* peut toujours servir de motif dans un fichier *plus grand*.

[1] Si j'ai bien compris mes collègues biologistes.