

TP Algorithmique du texte

1 Préambule

Choisissez un langage de programmation avec lequel vous êtes familier. Par défaut, et si vous ne savez pas quoi choisir, vous pouvez programmer en C/C++.

2 Manipulations de base

Exercice 1 *Écrire une fonction qui permet de rentrer deux mots u et v au clavier, et écrit les mots suivants (chacun sur une ligne) :*

1. uv
2. \bar{u} et \bar{v} (miroirs de u et v respectivement).
3. Le mot construit à partir de u et v en prenant la première lettre de u puis la première lettre de v , la deuxième lettre de u puis la deuxième lettre de v et ainsi de suite jusqu'à épuiser toutes les lettres de u et de v .

Exercice 2 *Complétez votre programme en donnant la possibilité d'ouvrir un fichier texte où sont stockés les deux mots, et d'enregistrer un fichier texte avec les résultats.*

3 Préfixes, suffixes

Exercice 3

1. *Écrire deux fonctions qui prennent deux mots en paramètre, et indiquent si le premier est un préfixe du second, et respectivement un suffixe du second. Ces fonctions renvoient un booléen.*
2. *Écrire une fonction qui renvoie tous les bords d'un mot passé en paramètre.*

4 Mots de Fibonacci

Exercice 4 *Écrire une fonction qui prend en paramètre un entier n et donne le n ème mot de Fibonacci (Indication : programmez en impératif ou en récursif, mais dans ce cas, attention à la mémoire. Vous pouvez ajouter des paramètres pour diminuer l'utilisation de la pile d'appels).*

5 Recherche naïve

Exercice 5 *Implémentez l'algorithme de recherche naïve d'un mot dans un texte. La sortie sera un fichier texte indiquant les positions trouvées des occurrences.*

Exercice 6 *Cherchez ou préparez des textes de plusieurs tailles, croissantes, et faites tourner votre programme dessus, en notant les temps d'exécution (la commande `time` peut vous aider.). Dressez un tableau donnant ces temps, et faites un graphique. Retrouvez la complexité théorique.*