



## INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR

### Algorithmique Avancée – Informatique et Télécommunication L3

#### Programmation Dynamique – Application de la distance de Levenshtein

Langage à utiliser : JAVA ou C#

- 1) Considérer la liste de mots en français : `liste_francais.txt`.  
(source <http://www.3zsoftware.com/fr/listes.php>)  
Observer le Format de ce fichier.  
Ecrire un code qui permet de charger les mots existants dans ce fichier dans une structure de données en mémoire vive (exemples : `ArrayList<>` et/ou `HashSet<>` en JAVA ou `List<>` et/ou `Dictionary<>` en C#).
- 2) Ecrire un sous-programme qui permet de tester si un mot donné par l'utilisateur existe dans le dictionnaire ou pas.
- 3) Ecrire un sous-programme qui permet de donner une liste de propositions de quelques mots les plus proches, selon la distance de Levenshtein dans le cas où un mot donné n'existe pas dans le dictionnaire.
- 4) Ecrire un sous-programme qui teste tous les mots (séparés par un caractère 'espace' ou autres séparateurs de votre choix) dans un fichier texte donné (par exemple : `texte.txt`) et produit une liste de corrections des éventuels mots erronés dans un autre fichier (par exemple `texte.err`) avec les propositions de remplacements, selon la distance de Levenshtein.

**Exemple :**

**Texte.txt**

Je suis trs en fomre

**Texte.err**

trs - très, tris, tas, trop.

fomre - forme, format.

- 5) Utiliser maintenant, le fichier `gutemberg.txt`. (source <http://www.3zsoftware.com/fr/listes.php>)  
Quelle est la différence entre `liste_francais.txt` et `gutemberg.txt` ?

Réviser vos réponses des questions 1) à 4) afin d'en optimiser le temps d'exécution avec le nouveau fichier.