

Contrôle continu LI 012
Devoir sur table n° 1
Aucun document autorisé.
Durée : 2 heures.

1. On suppose que l'on dispose des procédures et fonctions suivantes, pour la manipulation d'une liste de **caractères**, avec le fonctionnement décrit en cours.

```
fonction LONGUEUR(l : liste) : integer;  
    renvoie le nombre d'éléments de la liste  
fonction ELEMENT(p : integer, l : liste) : char;  
    renvoie le caractère qui se trouve à la position p  
procédure INSERER(x : char, p : integer, var l : liste);  
    insère le caractère x à la position p  
procédure SUPPRIMER(p : integer, var l : liste);  
    supprime l'élément qui se trouve à la position p
```

Écrire un algorithme qui, étant donnée une telle liste de caractères, affiche un par ligne les caractères de la liste, en commençant par la fin.

Cet algorithme sera écrit uniquement au moyen des primitives fournies. On supposera qu'il n'est pas nécessaire d'initialiser la liste.

N.B. : On ne demande pas la définition de ces procédures et fonctions !

2. On appelle *racinisation* (angl. *stemming*) un processus de découpage pseudo-morphologique de mots : par exemple, pour tout mot de longueur ≥ 5 , on décidera arbitrairement de considérer que le suffixe représente les 20% caractères les plus à droite.

Écrire un programme de racinisation : étant donné un tableau de N chaînes de caractères, le programme découpe tous les mots (de longueur ≥ 5) en créant deux tableaux : un pour les « racines », et l'autre pour les « suffixes ».

On veillera à ce qu'une racine ou un suffixe donné n'apparaisse pas plusieurs fois dans le tableau correspondant — sans pour autant que ces tableaux ne soient triés.

3. Étant donné un tableau de réels, non trié, on demande d'écrire un programme qui crée un second tableau, de même dimension, et le remplit avec les valeurs du premier tableau, de telle sorte que ce second tableau soit trié par ordre croissant. Le tableau initial ne doit pas être modifié.
-