Type de document	Rappels de 1re	Classe	Tle	Durée	2h	Date	16/09/2024
Thème et contenu(s)	Structures linéaires de données : les listes						
Capacités attendues	Savoir parcourir et manipuler une liste – comprendre les différences et les similitudes avec une chaîne de caractères (notion d'immutabilité)						
Prérequis	Programme de classe de 1re dont : structures conditionnelles, boucles						
Description	Exercices de révision un peu plus avancés autour des listes et des chaînes de caractères						

I) Réinventons la roue !

- Créer une fonction longueur(l) qui prend en paramètre une liste l (même vide) vide et renvoie sa longueur sous forme d'entier. Quelle fonction avons-nous redéfini ?
- Créer une fonction trouveIndice(el, l) qui prend en paramètre un élément el et une liste l et qui renvoie un entier correspondant à l'indice de la première occurrence de el dans l (-1 si el n'est pas dans l). De quelle méthode s'inspire notre fonction ?
- Quel problème avons-nous pour redéfinir la méthode pop() ? La fonction del() ?

II) Le loucherbem, une forme avancée de largonji

Le XIXème siècle voit l'argot de plus en plus utilisé dans la langue française écrite. Une de ses manifestations les plus emblématiques (encore présente aujourd'hui dans le langage argotique : larfeuille, loufoque…) est le Loucherbem, l'argot des bouchers. L'idée est la suivante : il faut cacher les mots existants en leur faisant subir des ajouts et des transformations. Le but de cet exercice est de transformer n'importe quel mot en son équivalent Loucherbem : c'est un exercice de chiffrement.
Pour parler Loucherbem :

1. Prendre la première consonne (ou les premières consonnes du mot) et les mettre à la fin du mot
2. À la place (donc au début du mot), mettre la lettre « l »
3. Ajouter un suffixe : dans un premier temps on ajoutera seulement le suffixe « **-em** »

Par exemple, le mot **boucher** peut être transformé comme suit :

1. On prend **b** qu'on rejette en fin de mot : _oucherb
2. On met à sa place la lettre **l** : **l**oucherb
3. On rajoute le suffixe **em** : loucherb**em**

II - a) Échauffement

Créer une fonction loucherbem(mot) qui transforme un mot en son équivalent loucherbem (renvoie le mot). On se limitera aux mots commençant par une seule consonne

II - b) Groupes de consonnes

Généraliser ça aux mots commençant par plusieurs consonnes. On pourra créer une fonction auxiliaire.

II - c) Gestion des consonnes finales

Gérer les cas où la première lettre rejetée en fin de mot se télescope avec une lettre muette en fin de mot initial. Par exemple, **gigot** doit devenir **ligogem** et non **ligotgem**

II - d) Variations

Au lieu d'utiliser toujours le suffixe **–em**, faire choisir la fonction aléatoirement entre un des suffixes suivants : **-em**, **-uche**, **-es**, **-ji**, **-oc**, **-ic**