

Devoir surveillé ☐

Examen ☒

Session: principale ☐
de contrôle ☒

Matière : Algorithmique

Enseignant(s) : Riadh ROBBANA

Filière(s) : MPI

Barème : Ex 1 : 8 pts, Ex 2 : 12 pts

Nombre de pages : Une page

Semestre: Deuxième

Date: juin 2014

Durée: 1h 30 mn

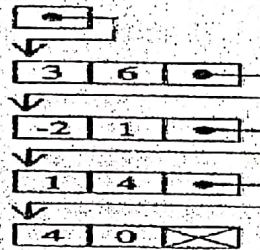
Documents: autorisés ☐
non autorisés ☒

Exercice 1 : Polynôme

Ecrivez un programme effectuant la lecture d'un polynôme et sa représentation en machine sous la forme d'une liste chaînée dont chaque maillon représente un monôme.

Par exemple, la figure ci-contre est une représentation possible du polynôme

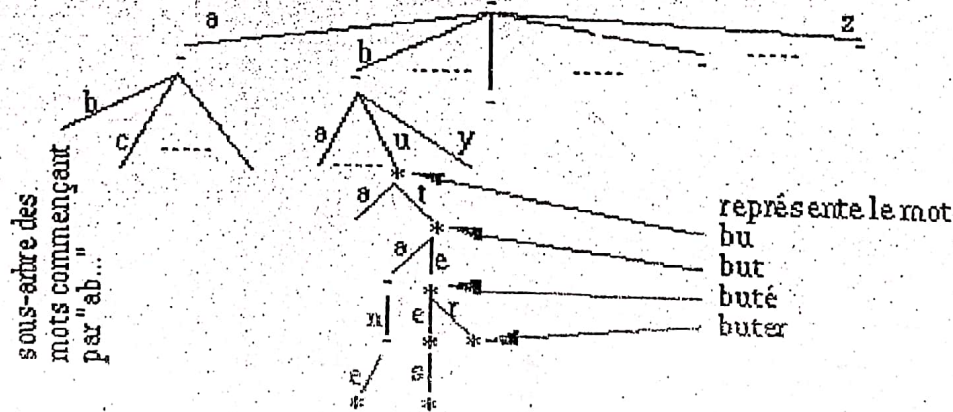
$$3 \cdot X^6 - 2 \cdot X + 1 \cdot X^4 + 4 \cdot X^0$$



- 1- Définissez la structure de donnée qui permet de représenter le polynôme.
- 2- Ecrivez une fonction récursive qui permet la saisie du polynôme.
- 3- Ecrivez une fonction récursive qui permet l'évaluation du polynôme pour une valeur de x donnée.

Exercice 2 : Arbre lexicographique

On construit un arbre multiple d'ordre 26 permettant de stocker des mots d'un dictionnaire de telle sorte que les chemins partant de la racine représentent des mots du dictionnaire, de la façon suivante :



Les arcs sont étiquetés avec les lettres de l'alphabet (on suppose ici que l'on ne tient pas compte des accents sur les lettres) et les nœuds contiennent un booléen indiquant si le chemin de la racine à ce nœud représente un mot complet ou pas (sur le dessin - = faux et * = vrai).

1. Donner en C la structure adéquate pour la représentation d'un tel arbre.
2. Ecrire une fonction Lecture qui construira la structure du dictionnaire à partir d'un texte en entrée qui se termine par le caractère # où tous les mots sont fournis.
3. Ecrire une fonction NbMots comptant les mots du dictionnaire ayant une longueur donnée. Si la longueur est 0, il faut retourner le nombre total de mots du dictionnaire, quelle que soit leur longueur.
4. Ecrire une fonction Trouve qui permet de savoir si un mot donné fait partie de l'arbre ou non.