Université de Batna 2
Département Informatique
Module : Algorithmique et structures de données

Série II : Pointeurs, variables dynamiques et listes chaînées

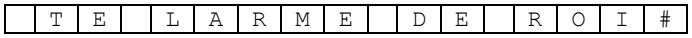
Exercice 1 Considérons une machine séquentielle de caractères. On définit un mot comme un ensemble de 12 caractères au maximum, précédés et suivis de caractères blancs, exception faite du premier mot du ruban qui peut ne pas être précédé de blancs, et du dernier mot du ruban qui peut ne pas être suivi de blancs. La fin des données du ruban est indiquée par le caractère #.

Soit la structure de données suivante :

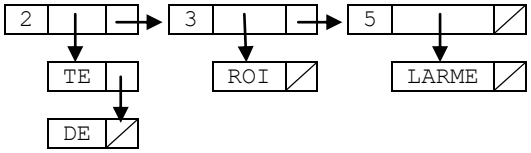
```
typedef struct wordNode* pWord;
struct wordNode {
       string word;
       pWord next;
};

typedef struct listNode* pList;
struct listNode {
       int size;
       pWord words;
       pList next;
};
```

Cette structure de données représente une liste de listes. La liste principale représente l'ensemble des tailles de mots trouvées sur le ruban. Cette liste doit être triée selon l'ordre croissant des tailles. Les nœuds de chaque sous-liste contiennent les mots ayant la taille du nœud correspondant dans la liste principale.

Exemple : avec le ruban suivant :



On obtient :



Écrire la fonction qui parcourt le ruban et construit la liste principale ainsi que ses sous-listes.

```
void createListOfWords(pList& head, char t[]);
```

Exercice 2

- Inversion d'une liste chaînée.
- Éclatement d'une liste chaînée en 2 listes (selon un critère à définir).

Exercice 3

On considère une liste bidirectionnelle circulaire de caractères. Écrire une fonction récursive logique qui teste si la liste représente un palindrome.

Problème

Soient les polynômes à exposants entiers et à coefficients réels.
$a_1x^{e1} + a_2x^{e2} + a_3x^{e3} + \ldots + a_n$
où $a_1, a_2, \ldots, a_n$ sont des réels.
$e_1, e_2, \ldots, e_n$ sont des entiers tels que $e_1 > e_2 > \ldots > e_n >= 0$

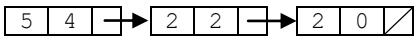
Pour représenter ces polynômes, on utilise des listes chaînées :
- Ces listes doivent être triées dans l'ordre décroissant des exposants.
- Les éléments dont les coefficients sont nuls ne doivent pas être représentés.

Structure de données :

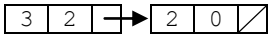
```
typedef struct node* ptr;
struct node {
       double coef;
       int exp;
       ptr next;
};
```

Exemple

$5x^4 + 2x^2 + 2$ sera représenté par :



$3x^2 + 2$ sera représenté par :



Écrire les sous-programmes suivants :

a) `int degree(ptr p);`
`// Cette fonction retourne le degré du polynôme p.`

b) `double coefficient(ptr p, int exp);`
`// Cette fonction retourne le coefficient associé à l'exposant exp dans le polynôme p.`

c) `void add(ptr p, ptr q, ptr& result);`
`// cette fonction réalise l'addition des polynômes p et q dans un polynôme à créer 'result'.`

d) `void multiply(ptr p, ptr q, ptr& result);`
`// Cette fonction réalise la multiplication des polynômes p et q dans un polynôme à créer 'result'.`

e) `void edit(ptr& p, int exp, double coef);`
`// Cette fonction affecte le coefficient coef à l'élément d'exposant exp dans le polynôme p.`

f) `void integrate(ptr p, ptr& result);`
`// Cette fonction calcule l'intégrale du polynôme p dans un polynôme à créer 'result'.`