

**ISIMA 1ère ANNEE**  
**STRUCTURES DE DONNEES ET ALGORITHMIQUE**

**Partiel 1 – décembre 2012**

**Durée : 2 heures**

**Documents autorisés : une feuille A4 manuscrite.**

**PROBLEME 1 (6 pts)**

Une chaîne de caractères contenant, entre autres, des suites de zéros et d'espaces est stockée dans une liste contiguë à raison d'un caractère par mot de la liste. La fin de la chaîne est signalée par la valeur NIL. Pour réduire la longueur de la liste, on effectue une compression de la chaîne en remplaçant une suite de k zéros par le caractère @ suivi de la valeur k, et une suite de k espaces par le caractère # suivi de la valeur k.

Par exemple la liste : A B 0 0 0 3 X □ □ □ □ Nil (le □ représente un espace)

Devient : A B @ 3 3 X # 4 Nil

La compression d'une suite de zéros ou d'espaces n'est réalisée que si elle permet un gain de place.

Écrire le programme de compression (sans créer de nouvelle liste).

**PROBLEME 2 (14 pts)**

Nous considérons des informations nécessitant un bloc de deux mots, l'un stockant une date représentée par un entier, l'autre un numéro d'identifiant. Ces informations sont ordonnées dans une liste dans le sens croissant des dates. Deux blocs successifs doivent avoir des dates différentes. Ainsi une telle liste peut représenter la succession d'événements (changements de valeur) d'un signal numérique et peut être utilisée dans un simulateur par exemple.

On se propose d'étudier l'insertion d'un nouvel événement (d,v) dans une liste déjà bien formée ou vide. Il faut placer le couple à la bonne place, puis s'il y a des événements après la position d'insertion, il faut tous les supprimer (ils deviennent périmés). S'il existe déjà un événement à la date d, il est remplacé par le nouvel événement (d,v).

1. Ecrire l'algorithme d'insertion d'un couple (d,v) dans le cas où la liste est contiguë avec pointeur de tête (m(t)) et pointeur de fin (m(f)). Ne pas libérer l'espace mémoire des événements périmés supprimés. On suppose que l'on a assez d'espace mémoire dans la liste après l'élément pointé par m(f).

2. Ecrire l'algorithme d'insertion d'un couple (d,v) dans le cas où la liste est implémentée par une liste chaînée. Libérer l'espace mémoire des éventuels événements périmés supprimés.

3. Laquelle des deux solutions semble la mieux adaptée dans les 3 cas suivants, justifier la réponse :

- 1- la liste peut devenir très longue ;
- 2- il y a souvent de nombreux éléments supprimés car périmés ;
- 3- de nombreux accès sont effectués dans la liste mais sa taille est peu importante car on supprime souvent les éléments de tête.

**NB : les algorithmes seront itératifs, écrits avec le langage utilisé en cours, accompagnés d'un schéma de la structure (ou d'un exemple) et commentés (au choix : principe ou commentaires dans le code). Donner en particulier la signification de toutes les adresses auxiliaires utilisées.**

**L'utilisation de procédures et fonctions est vivement recommandée. Si des procédures et fonctions du cours sont utilisées il faut les recopier. La trace n'est pas demandée.**