Examen de la session principale

**Matière :** STRUCTURES DE DONNEES

| | |
|---|---|
| **Filière / Classe : 1ère année Ingénieur Info** | **Documents : Non aut.** |
| **Date : 23/05/2019** | **Calculatrice : Non aut.** |
| **Durée : 1h30** | **Nb de pages : 2** |

***N.B. : Les réponses doivent être précises et claires***
***Bonne chance et bon travail***

## Exercice 1 : Les Files

On considère une file abstraite de caractères ne contenant que des S (déplacement Sud) et des N (déplacement Nord). De façon évidente un déplacement S est compensé par un déplacement N et réciproquement.

A partir des primitives de la structure file (vues en cours) :

Ecrire une fonction ***simplificationNS*(**var F: File**)**: **entier** qui prend en entrée une telle file et simplifie son contenu en produisant toutes les compensations possibles. A la fin il ne restera dans la file que les éléments qui n'ont pas pu être compensés (que des N ou que des S). La fonction retournera le nombre de compensations réalisées lors du traitement de la file.

Par exemple :

Si en entrée F= ]S,S,S,N,N,S,N,N,S,S,S,N,S[

On aura en sortie : F=] S,S,S [ et la fonction retournera 5.

## Exercice 2 : Liste chaînée

On se propose de compresser une liste chaînée triée. La compression se fera en remplaçant chaque séquence du même élément par un seul nœud contenant l'élément et son nombre d'apparition. Un seul parcours de la liste à compresser est permis.

1. Définir les structures de donnée appropriées pour représenter :
   a. la liste initiale décompressée **Liste_D**
   b. la liste compressée **Liste_C**


45, rue des entrepreneurs Charguia 2, Tunis Carthage 2035. Tél/fax 71 941 579

2. Ecrire la fonction **Compression** qui permet de compresser une liste
3. Ecrire la fonction **Décompression** qui, à partir des la liste compressée récupère la liste décompressée.

## Exercice 3 : Arbres Binaires de Recherche

Dans cet exercice, l'argument A est un arbre binaire de recherche dont les éléments sont des entiers :

1. Ecrire une fonction récursive ***NombreGaucheDroite(A)*** prenant en entrée un arbre binaire et donnant le nombre de noeuds de l'arbre qui ont à la fois un fils gauche et un fils droite.

2. Ecrire une fonction récursive ***Taille(A)*** donnant le nombre d'éléments contenus dans l'arbre.

3. Ecrire une fonction récursive ***Somme(A)*** prenant un arbre binaire dont les éléments sont des entiers et rendant la somme de ses éléments.

4. Ecrire une fonction ***moyenne(A)*** calculant la moyenne des éléments de l'arbre (supposé non vide).

5. Ecrire une fonction ***nbSommetsInternes(A)*** qui renvoie le nombre de sommets internes (autres que les feuilles) d'un arbre binaire.


45, rue des entrepreneurs Charguia 2, Tunis Carthage 2035. Tél/fax 71 941 579