المعهد الوطني للعلوم التطبيقية والتكنولوجيا
insat

Devoir surveillé ☐	Examen ☒	Session: principale ☒ de contrôle ☐
Matière : **Algorithmique 2**		Semestre: Deuxième
Enseignant(s) : Riadh ROBBANA		Date: mai 2016
Filière(s) : **MPI**		Durée: 1h 30 mn
Barème : Ex 1 : 6 pts, Ex 2 : 6 pts et Ex 3 : 8 pts		Documents: autorisés ☐ non autorisés ☒
Nombre de pages : Une page		

Exercice n°1 : On désire introduire une notion de priorité dans la file d'attente. La priorité est codée par un nombre entier positif, 0 étant le moins prioritaire. Lorsque l'on retire un élément de la file, on veut obtenir le plus ancien avec la priorité la plus haute. Lors du dépôt d'un élément on indique sa priorité :

1- Définir un nouveau type *fileprio* pour gérer la file avec priorités.

2- Ecrire les fonctions de dépôt et de retrait. Le dépôt correspond à l'opération Enfiler et le retrait correspond à l'opération Defiler.

Exercice 2 :

Soit A un arbre binaire

a - Ecrire une fonction itérative detruit_arbre qui libère la mémoire occupée par tous les éléments (nœuds et feuilles) de l'arbre A

b - Ecrire une fonction recursive detruit_arbre qui libère la mémoire occupée par tous les éléments (nœuds et feuilles) de l'arbre A

Exercice 3 :

On veut écrire une fonction de partition d'un arbre binaire de recherche autour d'un élément. Plus précisément, étant donné :

– un arbre binaire de recherche A

– et un élément x de clé c.

On veut obtenir deux ABR, G et D, tels que

– G contient tous les éléments de A de clé inférieure ou égal à c

– et D tous les éléments de A de clé supérieure strictement à c.

On veut que la fonction de partition fasse le moins d'opérations possibles. On évitera en particulier l'algorithme consistant à parcourir la liste de tous les éléments de A.

Pour simplifier l'écriture en C, on peut considérer que la fonction reçoit A ainsi qu'un arbre binaire B à un seul nœud contenant l'élément x et rend son résultat en faisant de G le sous arbre gauche de B et de D le sous-arbre droit de B (remarque : l'arbre B obtenu est un arbre binaire de recherche dont les éléments sont ceux de A plus l'élément x, placé à la racine).

1. Décrire une fonction en C de partition, l'expliquer sur un exemple.
2. Donner la complexité de l'algorithme de partition.