

Examen – Session de rattrapage

Matière : Atelier programmation II
Enseignant : M. Majdi JRIBI
Filière : MPI
Nombre de pages : 4 pages

Semestre: Second semestre
Date: 24 Juin 2015
Durée: 1h00
Documents : non autorisés

Les réponses doivent être rédigées obligatoirement sur les feuilles de réponse (pages 3 et 4)
L'examen contient 4 pages. Seulement les pages 3 et 4 sont à rendre.

Exercice 1 : Arbre binaire et arbre binaire de recherche

Considérons les déclarations suivantes :

```
typedef struct arb
```

```
{     int val;  
      struct arb *fg;  
      struct arb *fd;  
} arb;
```

```
typedef arb* arbre;
```

- 1- Ecrire en langage C une fonction récursive *void detruire_arbre (arbre ar)* qui prend en entrée un arbre binaire et qui permet de libérer la mémoire occupée par tous les nœuds de cet arbre.
- 2- Ecrire en langage C une fonction *int est_binaire_recherche(arbre ar₁)* qui prend en entrée un arbre binaire et qui renvoie la valeur 1 si l'arbre ar₁ est binaire de recherche sinon elle renvoie la valeur 0.
- 3- Ecrire en langage C une fonction récursive *int compare (arbre ar₁ , arbre ar₂)* qui prend en entrée deux arbres binaires et qui renvoie la valeur 1 si les deux arbres ar₁ et ar₂ sont identiques sinon elle renvoie la valeur 0.

Exercice 2 : les files

Considérons les déclarations suivantes d'une file d'entiers:

```
typedef struct Noeud
{
    int valeur;
    struct Noeud * suivant;
} Noeud;
```

```
Typedef Noeud * file;
```

Soient les fonctions prédéfinies suivantes concernant des traitements sur les files:

- *Tester si une file est vide : int estvide(file fil);* (Elle permet d'envoyer 0 si la file est vide).
- *Enfiler un élément dans une file : file enfiler(file fil, int val);* (Elle permet d'ajouter un entier à la queue de file).
- *Défiler un élément d'une file : file defiler(file fil);* (Elle permet de supprimer la tête de la file et rend la nouvelle file sans la tête).
- *Déterminer la tête de la file: int tete(file fil);* (Elle récupère la valeur de la tête de la file).

Soient F_1 et F_2 deux files d'entiers. Chaque file est triée par ordre croissant (l'élément le plus petit se trouve à la tête de la file).

En utilisant seulement les fonctions prédéfinies citées, écrire une fonction en langage C *file fusion (file F₁, file F₂)* qui permet de fusionner les deux files F_1 et F_2 dans la file F_2 de telle sorte que la file finale soit triée par ordre croissant (la tête de la file finale contient l'élément le plus petit).

Examen – Session de rattrapage

Matière : Atelier programmation II
Filière : MPI
Nom :
CIN :

Semestre: Second semestre
Date: 24 Juin 2015
Prénom :
Numéro Inscription :

Feuilles de réponse

Exercice 1 :

1)

void detruire_arbre (arbre ar)

2)

int est_binaire_recherche(arbre ar₁)

Ne rien écrire ici

Nom :
Prénom :
N° inscripti

Exercice 2 :

3)

int compare (arbre ar₁, arbre ar₂)

file fusion (file F₁, file F₂)