

Examen n°01 d'Algorithmique et Structures de données avancées

2^{ème} année licence
(01 h 30 min)

« La propreté de la feuille de réponses est exigée et notée (1 point) » 1

L'objectif de cet examen consiste à trier une liste de valeurs entières. Autrement dit, nous voulons ordonner les valeurs d'une liste linéaire chaînée selon l'ordre croissant, tout en ignorant les valeurs répétées.

Exemple :



Pour ce faire, nous allons diviser le travail en quatre parties :

Partie 1 : (Remplir la liste) (5 points)

1. Donner la définition, en langage algorithmique, d'un maillon dans une liste linéaire chaînée. (2 pts)

```
Type Maillon=Structure 0.5
    Valeur : Type_qq 0.5
    Suivant : Pointeur(Maillon) 1
Fin
```

2. Donner la déclaration, en langage algorithmique, de la tête de cette liste. (1 pts)

```
Var
    Tête : Pointeur(Maillon) 1
```

3. Ecrire, en langage algorithmique, la procédure qui permet de créer une liste à partir des valeurs entières fournies par l'utilisateur. (2 pts)

```
Procédure CréerListe(Var Tête : Pointeur(Maillon), N :entier)
Var
  P,Q : Pointeur(Maillon)
  I : Entier
Début
  Tête ← Nil
  Pour i allant de 1 à N faire
  Dpour
    Lire(Val)
    Allouer(P)
    Aff_Val(P,Val)
    Aff_Val(P,Nil)
    Si(Tête = Nil) Alors
      Tête ← P
    Sinon
      Aff_Adr(Q,P)
    Q ← P
  Fpour
Fin
```

Partie 2 : (Création d'un arbre binaire de recherche (ABR)) (8 points)

Dans cette partie, nous allons créer un arbre binaire de recherche à partir d'une liste linéaire chaînée.

1. Donner la définition, en langage algorithmique, d'un Nœud dans un arbre binaire de recherche. (1pts)

```
Type Nœud=Structure
  Valeur : Type_qq
  FG,FD : Pointeur(Nœud)
Fin
```

2. Donner la déclaration, en langage algorithmique, de la racine de cet arbre. (1pts)

```
Var
  Racine : Pointeur(Nœud)
```

3. Ecrire, en langage algorithmique, la fonction d'insertion d'une valeur dans un arbre binaire de recherche (3pts).

```
Procédure InsérerABR (Var Racine : Pointeur(Nœud), Val : entier)
Var
    N : Pointeur(Nœud) 0.25
Début
    Si (Racine = Nil) Alors 0.25
        Dsi
            Allouer(Racine) 0.25
            Aff_Val(Racine, Val)
            Aff_FG(Racine, Nil) 0.5
            Aff_FD(Racine, Nil)
        Fsi
    Sinon
        Dsn
            Si (Valeur(Racine) <> Val) Alors 0.25
                Dsi
                    Si (Valeur(Racine) > Val) Alors
                        InsérerABR(FG(Racine), Val) 0.75
                    Sinon
                        InsérerABR(FD(Racine), Val) 0.75
                Fsi
            Fsn
        Fsn
Fin
```

4. Ecrire, en langage algorithmique, la procédure qui permet de créer un arbre binaire de recherche à partir d'une liste linéaire chaînée (3pts).

```
Procédure CréerABR-Liste (Var Racine : Pointeur(Nœud), Tête : Pointeur(Maillon))
Var
    P : Pointeur(Maillon)
    Val : Entier 0.5
Début
    P ← Tête 0.25
    Racine ← Nil 0.25
    Tant que (P <> Nil) Faire 0.5
        Dtq
            Val ← Valeur(P) 0.5
            InsérerABR(Racine, Val) 0.5
            P ← Suivant(p) 0.5
        Ftq
    Fin
```

Partie 3 : (Affichage de l'arbre binaire de recherche) (4 points)

1. Quel est le type de parcours qui nous permet de visiter les valeurs de l'ABR selon l'ordre croissant. (1pt)
- a. Pré-Ordre
 - b. In-Ordre** 1
 - c. Poste-Ordre

2. En modifiant la procédure du parcours précédent (réponse de la question précédente), écrire une procédure, en langage algorithmique, qui permet de créer une liste triée (ordonnée) à partir de l'arbre binaire de recherche (3pts) (◆).

Procédure CréerListe-ABR (Racine : Pointeur(Noeud), var Tête : Pointeur(Maillon))

Début

Si (Racine <> Nil) Alors 0.75

Dsi

CréerListe-ABR (FG (Racine), Tête) 0.75

InsérerListeFin (Tête, Valeur (Racine)) 0.75

CréerListe-ABR (FD (Racine), Tête) 0.75

Fsi

Fin

◆ ***Vous pouvez utiliser directement (sans programmation) la procédure d'insertion à la fin d'une liste :***

Procédure InsérerListeFin (Var Tête : Pointeur (Maillon), Val : Entier)

Partie 4 : (Programme principal) (2 points)

En utilisant les modules (***CréerListe***, ***CréerABR-Liste***, ***CréerListe-ABR***), écrire le programme principal (Sans la partie déclaration de variable, fonctions ...etc.) qui permet de :

- 1) Créer une liste Linéaire chaînée.
- 2) A partir cette liste, créer un ABR.
- 3) Créer une liste ordonnée à partir de l'ABR formé.

{Partie déclaration de variables}
{Partie déclaration des fonctions/Procédures}

Début

Ecrire (" Donner le nombre de Valeurs ")

Lire (N) 0.5

CréerListe (Tête, N) 0.5

CréerABR-Liste (Racine, Tête) 0.5

Tête ← NULL (Ou bien Tête2) 0.5

CréerListe-ABR (Racine, Tête)

Fin

Bonne chance