

Correction du Test (Section A)

17/12/2016

- Exercice 1

Type ArbreNaire

Début

Clé : Caractère ;

Fils : *ArbreNaire ;

Frère : *ArbreNaire ;

Fin

Typedef struct ArbreNaire

{

char Clé;

struct ArbreNaire *Fils;

struct ArbreNaire *Frère;

} **ArbreNaire ;**

Procédure AfficheInverse (A : *ArbreNaire)

Début

Si (A <> Nil) Alors

 F : *File ; P : *Pile ; Tmp : *ArbreNaire ;

 Enfiler (F, A) ;

 Tant que (Non-Vide (F)) faire

 Tmp = Défiler (F);

 Empiler (P,Tmp);

 Si (Tmp.Fils <> Nil) Alors

 Tmp = Tmp.Fils ;

 Tant que (Tmp <> Nil) faire

 Enfiler (F, Tmp) ;

 Tmp= Tmp.Frère ;

 Fait ;

 Fsi ;

 Fait ;

 Tant que (Non-Vide (P)) faire

 Tmp = Dépiler (P) ;

 Ecrire (Tmp) ;

 Fait ;

 Libérer (F) ; Libérer (P) ;

Fsi;

Fin.

- **Exercice 2**

Type ArbreBinaire

Début

data : entier ;
FG : *ArbreBinaire ;
FD : *ArbreBinaire ;
Fin

Type Liste

Début

Elément : entier ;
Suivant : *Liste ;
Fin

Fonction SommeChemin (A : *ArbreBinaire, L : *Liste, S : entier) : *Liste
Début

Si (A <> Nil) Alors

S = S + A.data;

Si (A.FG = Nil et A.FD = Nil) Alors L = InsertListe (L, S);

Sinon

L = SommeChemin (A.FG, L, S);

L = SommeChemin (A.FD, L, S);

Fsi;

Fsi;

Retourner (L);

Fin.

Fonction InsertListe (L : *Liste, S : entier) : *Liste

Début

T : *Liste ; T = Allouer(Liste);

T.Elément = S ;

T.Suivant = L;

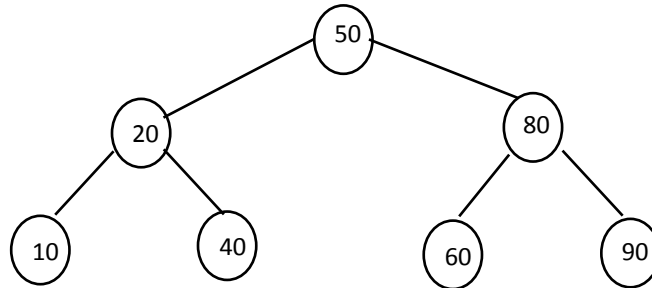
L = T;

Retourner (L);

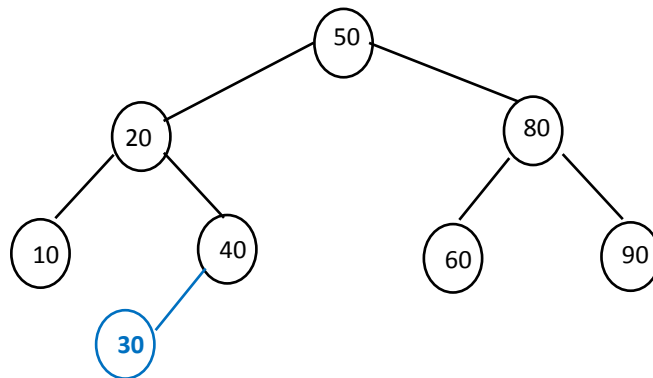
Fin.

- **Exercice 3**

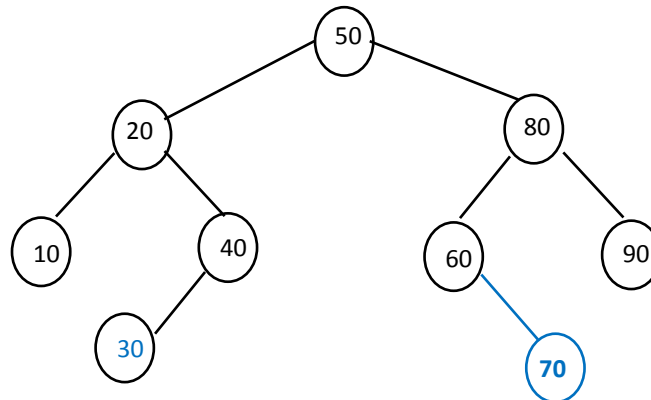
- **Arbre Binaire de Recherche Equilibré :**



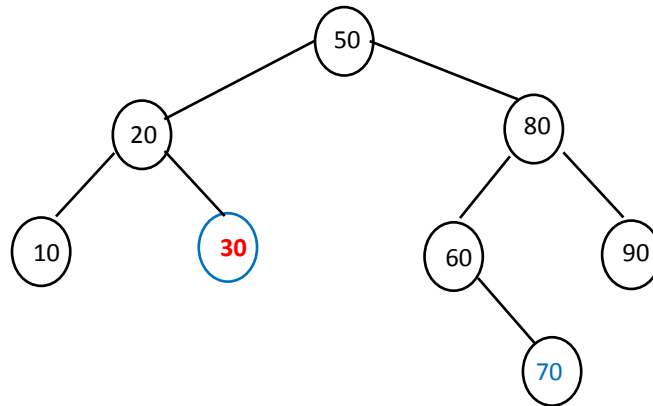
- **Insertion du nœud 30**



- **Insertion du nœud 70**



- **Suppression du nœud 40**



- **Suppression du nœud 80**

