UFAS 2012/2013
Faculté des Sciences
Département d'informatique
Module : ASD1
Année : 2ème Licence Académique

Epreuve de moyenne durée

Exercice 1(7pts)

Ecrire une fonction **Fusion (L_1, L_2)** qui reçoit deux listes chaînée simples d'entiers L_1 et L_2 triées dans l'ordre croissant, et retourne une liste chaînée triée dans le même ordre et contenant les deux listes.

Il faut déclarer la structure de la liste.

Aucun nouvel espace mémoire ne sera alloué dynamiquement.

Exercice 2 (7pts)

Ecrire une fonction **Inverse (L)** qui reçoit une liste chaînée simple d'entiers et retourne l'inverse de cette liste en utilisant une pile d'entiers.

Il faut déclarer la structure de la pile et réécrire ses fonctions de base : init(p), pilevide(p), empiler(p,val), depiler(p), et sommet(p)

Exercices 3 (6pts)

Un arbre d'arité 3 est un arbre dont chacun des nœuds possède au plus trois fils.

1. Déclarer une structure d'un arbre d'arité 3.
2. Ecrire une fonction qui prend un arbre d'arité 3 et retourne le nombre de feuilles dans cet arbre.

Correction

Exo1 :

```
struct element
{   int valeur ;
        Struct element *suivant ;
} ;
typedef  element *Telement
typedef  Telement *Tliste ;       /* -------------0.5pt-----structure---------*/

Tliste Fusion(Tliste L1, Tliste L2)/*-----------1pt-----------entête---------*/
{/*---------------- 5 pts pour le corps de la fonction----------------*/
tliste tmp, L3;

if (L1==NULL)/*------------0.5 pt pour la verification si une des 2 listes est vide*/
        return L2;
else {
       if (L2==NULL)
        return L1;
      else
       {/*-------------------------initialisation de L3------ 1pt-------*/
        if ((L1->valeur)<(L2->valeur))
             {L3=L1 ;L1=L1->suivant ;}
        else {L3=L2 ; L2=L2-suivant};

        tmp=L3 ;
        while ((L1 !=NULL)&&(L2 !=NULL) /*-------boucle 2.5 pt------*/
        {
              if( (L1->valeur)<(L2->valeur))
               {      tmp->suivant=L1 ;
                      tmp=L1 ;
                      L1=L1->suivant ;

                } ;
              else
              {       tmp->suivant=L2 ;
                      tmp=L2 ;
                      L2=L2->suivant ;
               };
        };
        /*--------le reste de l'une ou l'autre liste -------1pt------*/
        If (L1!=NULL)
              tmp>suivant=L1;
```

```
        if(L2!=NULL)
                tmp->suivant=L2;
};
return L3; /*-----0.5----------*/
};
```

Exo2

```
struct element
{   int valeur ;
    struct element *suivant ;
} ;

typedef  element *Telement
typedef  Telement *Pile ;----------------------------------0.5pts

// déclaration de toutes les fonctions de manipulation des piles//----------------1pts

Tliste Inverse (Tliste L) ------------------0.5
{ ----------------------4.5pts pour le corps de la fonction

Tpile p=init(p);   ------0.5 initialisation de la pile
tmp=L;

while (tmp!=NULL)----2 pts--------
{
         p=empiler(p,tmp->valeur);
        tmp=tmp->suivant ;
};

tmp=L;
while(tmp!=NULL)------2pts
{
        tmp->valeur=sommet(p);
        p=depiler(p);
        tmp=tmp->suivant ;
} ;
return L ;     -------------------------0.5
};
```

Exo3

Cette exercice a été traité en cours pour un arbre binaire.
Si l'étudiant reprends sans aucune faute la fonction pour un arbre binaire, je propose de donner 2/6.

La solution pour un arbre binaire est ce qui écrit en noir (en enlevant ce qui est écrit en rouge)

a)

```
struct element
{       int valeur ;
        struct element *filsgauche ;
        struct element *filscentre ;
        struct element *filsdroit ;
} ;

typedef  element *Telement
typedef  Telement *Tarbre ;----------1.5 pour la structure
```

b)

int nbfeuilles(Tarbre arbre) ----------0.5 pour l'entête----------

----------4 pts pour le corps de la fonction----------

```
{
If (arbre==NULL) return 0 ;// en cours nous avons utliser une fonction estvide(arbre).
if( (arbre->filsgauche==NULL)&&(arbre->filscentre==NULL)
      &&(arbre->droit==NULL))
  return 1 ; ----------2pts--------
else
 return (nbrefeuilles(arbre->filsgauche)+
         nbrefeuilles(arbre->filscentre)+
          nbrefeuilles(arbre->droit)) ;----------2pts pour le else ….

} ;
```