TELECOM Lille1 **Avril 2012**

Nom : Prénom : N° : Durée : 2 heures

QROC Module A31TTN - Structures de données

Documents non autorisés, les réponses sont à donner sur la présente feuille.

Exercice 1 – *Récursivité (4 points)*

On considère les fonctions ***f*** et ***g*** suivantes :

`int f(int n, int a) {`
`  if (n<=1) {`
`  return a;`
`  } else {`
`  return f(n-1,n*a);`
`  }`
`}`
	`int g(int n) {`
	`  int a = 1;`
	`  while (n>1) {`
	`  a = n*a;`
	`  n = n-1;`
	`  }`
	`  return a;}`

1) Que vaut ***f(5,1)***, donner la suite d'appels générée par ***f***.

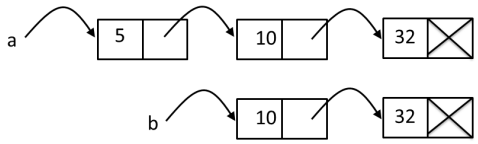
1) Que vaut ***g(5)***, donner la suite d'appels générée par ***g***.

Exercice 2 *– Listes chaînées (6 points)*

*On considère la déclaration de la structure de données ainsi que les fonctions **X**, **Y**, **Z** et **V** suivantes :*

```
typedef struct element
{
   int val;
   struct element *next;
}llist;
 typedef llist* list;
```

On considère aussi les listes suivantes :



1) Expliquer la fonction X, que retourne **X(a)**, donnez un nom significatif à la fonction **X**?

```
 int X(list a) {
   if(a == NULL)
      return 0;
   return X(a->next) + 1;
}
```

2) Expliquer la fonction Y (en illustrant par un schéma), que retourne **Y(b,5)**, donnez un nom significatif à la fonction **Y**?

```
list Y(list a, int value) {
   element* e = malloc(sizeof(element));
   e->value = value;
   e->next = a;
   return e;
}
```

3) Expliquer la fonction **Z** (en illustrant par un schéma), que retourne **Z(a,0)**, donnez un nom significatif à la fonction **Z**?

```
list Z(list a, int value) {
   element* e = malloc(sizeof(element));
   e->value = value;
   e->next = NULL;
   if(a == NULL)
      return e;
   else {
      element* t = a;
      while(t->next != NULL)
         t = t->next;
      t->next = e;
      return a;
   }
}
```

3) Expliquer la fonction **V** (en illustrant pas un schéma), que retourne **v(b,a)**, donnez un nom significatif à la fonction **V**?

```
list V(list a, list b) {
   if(a != NULL) {
      element* t = a;
      while(t->next != NULL)
         t = t->next;
      t->next = b;
      return a;
   } else
      return b;
}
```

<u>Exercice 3</u> *– Les arbres binaires (10 points)*

On considère la suite de nombres suivante : 88, 91, 38, 56, 50, 12, 2, 55, 83, 0.

1) Partant d'un arbre binaire de recherche vide, on ajoute successivement les valeurs de la suite ci-dessus. Donnez en justifiant votre raisonnement l'arbre binaire résultat *(il ne s'agit pas de donner la version équilibrée de l'arbre)*.

2) Vérifier si l'arbre obtenu est équilibré, s'il n'est pas équilibré, équilibrez-le (en recommençant de nouveau l'insertion de la séquence 88, 91, 38, 56, 50, 12, 2, 55, 83, 0). Expliquez les opérations de rotation effectuées.

3) On considère la structure de données suivante pour représenter cet arbre :

```
typedef struct Arbre {
        int Noeud;
        struct Arbre * SAG;
        struct Arbre * SAD;
} Arbre;
```

3.1) On dit qu'un arbre est ***tournoi*** si pour tout nœud, la valeur contenue dans ce nœud est inférieure à celles contenues dans ses fils. Écrire une fonction testant si un arbre est ***tournoi***.

3.2) On dit qu'un arbre est **listomorphe** si le fils gauche de tous ses nœuds est vide. Écrire une fonction qui teste si un arbre est ***listomorphe***.

3.3) Donner la fonction qui calcule la profondeur (hauteur) d'un arbre.

Bonne chance.