L3 - SRI
CC2 - Informatique
Décembre 2018 – 1 heure 30
(documents autorisés)

Pour être comptabilisée, toute réponse devra être justifiée.

1 Compréhension d'un programme C (10 points)

Soit le programme C suivant :

```
// Types
// ////////////////////////
typedef struct etiq1 {
   int champ1 ;
   struct etiq1 * champ2 ;
   struct etiq1 * champ3 ;
   } Type1 ;
typedef Type1 * Type2;

// Fonctions
// ////////////////////////
void fonction1Type2(Type2 o) {
   *o = NULL ;
   }
Type2 fonction2Type2(int x) {
   Type2 o2 = (Type2)malloc(sizeof(Type1)) ;
   assert(o2 != NULL) ;
   o2.champ1 = x ;
   o2.champ2 = o2.champ3 = NULL ;
   return o2 ;
   }
int fonction3Type2(Type2 o) {
   return (o == NULL) ;
   }
Type2 fonction4Type2(Type2 o, int x) {
   if (fonction3Type2(o))
       return fonction2Type2(x) ;
   if (o->champ1 >= x)
       o->champ2 = fonction4Type2(o->champ2,x) ;
   else
       o->champ3 = fonction4Type2(o->champ3,x) ;
   return o ;
   }
void fonction5Type2(Type2 o) {
       fonction5Type2(o->champ2) ;
       printf("%d ",o->champ1) ;
       fonction5Type2(o->champ3) ;
   }
```

```
// Programme de test
// ////////////////////////
int main () {
  Type2 o ;

  fonction1Type2(&o) ;

  o = fonction4Type2(o,10);
  o = fonction4Type2(o,2);
  o = fonction4Type2(o,30);
  o = fonction4Type2(o,4);

  fonction5Type2(o) ;
}
```

1. Des erreurs se sont glissées dans ce programme (soit des erreurs provoquant une erreur de compilation, soit des erreurs risquant de provoquer une erreur à l'exécution). Trouvez-les et expliquez comment les corriger.
2. Une fois le programme corrigé, décrivez son exécution et donnez les résultats obtenus.
3. En vous inspirant des structures de données vues en TD et en TP, dites ce que représentent les `Type1` et `Type2`. Proposez alors des noms plus explicites pour chaque type, chaque champ de structure et chaque fonction.
4. Y-a-t-il parmi les fonctions données des fonctions récursives ? Si oui, précisez de quel type de récursivité il s'agit (primitive ou pas, terminale ou pas, transformable terminale ou pas).

2 Problème (10 points)

On veut faire l'implémentation en C de l'algorithme de Dijkstra pour un graphe orienté dont les arcs sont pondérés avec des entiers. Pour cela, nous avons besoin d'implémenter une liste dynamique simplement chaînée triée. Votre travail sera le suivant :

1. Proposez un type en C pour implémenter cette liste (ce type sera noté `Liste`).
2. Donnez le code C des fonctions suivantes (vous respecterez *scrupuleusement* les prototypes indiqués) :
   — `Liste initListe()` : fonction permettant d'initialiser une liste vide,
   — `Liste ajoutDansListe(Liste l, int x)` : fonction permettant d'ajouter la valeur `x` à la liste `l` de telle sorte que le résultat soit toujours une liste triée par ordre croissant,
   — `void afficheListeCroissant(Liste l)` : fonction permettant d'afficher le contenu d'une liste de la plus petite valeur jusqu'à la plus grande.
3. Sachant qu'il pourrait être intéressant d'avoir les valeurs dans l'odre décroissant, que vous faut-il modifier pour obtenir une fonction `void afficheListeDecroissant(Liste l)` efficace (c'est-à-dire qui fait l'affichage avec un seul parcours de la liste). Indice : exploiter la notion de récursivité.
4. L'utilisateur de votre liste va maintenant écrire son code pour l'algorithme de Dijkstra. Que devez-vous lui passer pour qu'il puisse travailler mais sans savoir comment vous avez codé la liste. Donnez l'architecture des fichiers et décrivez brièvement ce qu'on doit y trouver.