

Cartouche du document

Année : ING 1

Matière : Programmation C

Activité : Examen

Objectifs

Cet examen teste vos aptitudes en programmation en C

- à traduire en C des algorithmes
- à corriger les erreurs d'un source C
- à manipuler les pointeurs
- à utiliser la gestion bufferisée de fichiers en C

Tous les documents (electronique et autres) sont autorisés.

La durée de l'examen est de 1h20 heures.

L'examen est assez long. Vous êtes donc notés sur 24.

Sommaire des exercices

- 1 - Tri fusion (7pts)
- 2 - Les pointeurs (7pts)
- 3 - Quelques erreurs de compilation (4pts)
- 4 - Les fichiers (6pts)

Corps des exercices

1 - Tri fusion (7pts)

Enoncé :

Soit le type abstrait Liste vu en TD, sur lequel nous ajoutons l'opération d'extension réalisant l'algorithme de Tri Fusion ci dessous:

```
//L'opération FusionnerListesTriees a été vue et corrigée en TD d'algo
Constructeur Liste: triFusion (Liste l) : Liste
Références locales
Liste l1,l2
```

Debut

```
l1 <- listeVide()
l2 <- listeVide()
Si split(L1,L2) Alors
  Debut
```

```
    retourner fusionnerListesTriees(triFusion(l1),trifusion(l2))
```

```
Fin
Sinon
Debut
    retourner l
Fin
Fin

//On découpe la liste observée en deux sous listes
//contenant chacune la moitié des éléments (1 sur 2).
//L1 et L2 sont deux paramètres modifiés par cette opération
//L'opération retourne faux si la liste de départ est vide
Observateur Liste: split(Liste l1,Liste l2) : Booleen
Observé l

Références locales
Liste ltemp
Element e

Debut
    ltemp <- l
    Si estVide(ltemp) Alors
        Debut
            retourner Faux()
        Fin
    Sinon
        Debut
            Faire
                Debut
                    e <- premier(Ltemp)
                    ajouter(l1,e)
                    ltemp <- reste(ltemp)
                    Si Non estVide(Ltemp) Alors
                        Debut
                            e <- premier(ltemp)
                            ajouter(l2,e)
                        Fin
                Fin
            Tant que Non estVide(ltemp)
        Fin
    Retourner Vrai()
Fin
```

Question 1)

Enoncé de la question

Implémenter en C, les opérations triFusion, split, fusionListesTriees. On pourra réutiliser des sources .h et .c faits en TDs et édités sur le site Arel.

On précisera bien ce qui doit être mis dans les fichiers d'entête (.h) et les fichiers sources (.c).

2 - Les pointeurs (7pts)

Enoncé :

On désire implémenter en C, une liste linéaire doublement chaînée (suivant et précédent). Pour cela on utilisera les types suivants :

```
struct SCell
{
    struct SCell * pprec;
```

```
void * pinfo;
struct SCell * psuiv;
} SCell;

typedef struct SCell * Cellule;

struct
{
    Cellule premiereCellule;
} SListe;

typedef struct SListe * Liste;
```

Question 1)

Enoncé de la question

Ecrire en C, la fonction qui permet de créer une liste linéaire doublement chaînée vide.

Question 2)

Enoncé de la question

Ecrire en C, la fonction qui permet d'ajouter un élément en début de liste.

Question 3)

Enoncé de la question

Ecrire en C, la fonction qui permet d'insérer un élément à une place donnée en paramètre sous forme d'un entier (On indicera la liste à partir de 0 pour la première cellule).

Question 4)

Enoncé de la question

Ecrire en C, les fonctions `resteAvant` et `resteAprès`. Ces deux fonctions sont les équivalentes de `listeReste` mais en tenant compte du double chaînage.

3 - Quelques erreurs de compilation (4pts)

Enoncé :

Corriger les erreurs de ce source (warning et syntax error). On utilisera la commande `gcc` avec l'option `-c` pour s'arrêter à la traduction du code : `gcc -c ??? .c`.

```
int main()
{
    float res1, res2;

    res = 0;
    for(i = 0; i < 10, i++)
    {
        res1 = res1 + i
        res2 = res2 + i * i;
    }
    res1 /= 10;
    res2 = res2 /10 - res1 * res1;
    afficherIntervalle(res1,res2)
}

void afficherIntervalle(float res1, float res2)
{
```

```
} printf("[%f, %f]\n", res1 - sqrt(res2), res1 + sqrt(res2))
```

4 - Les fichiers (6pts)

Enoncé :

Dans cet exercice, on manipule un fichier qui contient dans l'ordre en non formaté :

- deux entiers n1 et n2
- n1 nombres réels
- n2 caractères

Question 1)

Enoncé de la question

Ecrire en C, une fonction qui reçoit le nom d'un tel fichier et qui affiche tous les réels puis tous les caractères contenus dans ce fichier.