Université Libanaise
Faculté des Sciences 1

I2204 - INFO216 F

Programmation Impérative II

Session2 2016-2017
15 Julliet 2017
Durée: 2 heures

Exercice I [30 min, 25 points]

Voici un petit programme qui a des problèmes (6 pour être exact). Vous devriez trouver tous les bugs et donner une brève explication de chaque problème. Vous ne devez pas réparer les bugs (et il se peut qu'il n'y ait pas de corrections évidentes dans certains cas), mais parfois un moyen simple d'expliquer un bug est de montrer comment le corriger.

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   typedef struct point {
4       int x, y, z;
5   } Point, *PointPtr;
6   PointPtr add(PointPtr a, PointPtr b) {
7       Point res;
8       res.x = a->x + b->x;
9       res.y = a->y + b->y;
10      res.z = a->z + b->z;
11      return &res;
12  }
13  int main(int argc, char* argv[]) {
14      Point a = {1, 2, 3};
15      PointPtr b;
16      b->x = 3;
17      b->y = 5;
18      b->z = 7;
19      PointPtr c = malloc(sizeof(PointPtr));
20      c = add(&a, b);
21      printf("a = (%d, %d, %d)\n", a.x, a.y, a.z);
22      printf("b = (%d, %d, %d)\n", b->x, b->y, b->z);
23      printf("a + b = (%d, %d, %d)\n", c->x, c->y, c->z);
24      free(b);
25      free(c);
26  }
```

Pour répondre, copiez et remplissez le tableau suivant :

no bug	no ligne	bug	explication
1			
2			
3			
4			
5			
6			

Exercice II [30 min, 25 points]

Considérez le type de données suivant :

```
typedef struct noeud{int d; struct noeud *suiv, *prec;} noeud;
```

Écrivez une fonction "inverser" qui inverse l'ordre des noeuds dans une liste doublement chainée d'entiers donnée. Par exemple, la liste [3⇌9 ⇌5⇌1] devient après l'appel à inverser [1⇌ 5⇌ 9⇌ 3].

Exercice III [60 min, 45 points]

Nous voulons créer un index pour un texte donné. Nous allons le faire étape par étape, en suivant les questions 1 à 5. Tout d'abord, nous écrivons une fonction "creerTabLC" qui prend comme paramètre un nom de fichier texte, crée et retourne un tableau de 26 listes chaînées triées, une pour chaque lettre de l'alphabet. La fonction devrait lire le fichier texte et insérer chaque mot dans une des listes en fonction de sa première lettre, par exemple: tout mot qui commence par 'a' va, trié, à la liste chainée à l'entrée 0 du tableau. Deuxièmement, nous écrirons la fonction "ecrireIndex" qui utilise l'index construit dynamiquement pour écrire un nouveau fichier texte contenant l'index.
Notez qu'il n'y aura pas de doublons dans les listes et que la comparaison devrait être insensible à la casse. Notez également que les ponctuations et les guillemets au début et à la fin des mots ne doivent pas être comptabilisées.

1. Définissez la structure appropriée "nœud" pour contenir un mot et un champ "suivant".
2. Écrivez la fonction "creerTaby26" qui alloue dans la mémoire dynamique un tableau de 26 pointeurs de noeud, les initialise à NUL et renvoie l'adresse du tableau.
3. Ecrivez la fonction "insererTrier" qui insère un mot donné à son bon endroit dans une liste chainée donnée. La fonction ne doit rien faire si le mot existe déjà dans la liste. La comparaison doit être insensible à la casse.
4. En utilisant les fonctions des questions 2 et 3, écrivez la fonction "creerTabLC". Exemple: si vous recevez le nom du fichier "in.txt" dans la fig. 1 ci-dessous, la fonction devrait construire et remplir l'index dans la mémoire dynamique, fig. 2, puis le retourner.
5. Écrivez la fonction "ecrireIndex" qui, étant donné un index, copie son contenu (mots) dans un fichier texte. Chaque ensemble de mots correspondants à une letter, sera sur une ligne, séparé par des tirets, la ligne commence par la lettre de l'index suivie de deux points verticaux (:). À titre d'exemple, si l'on passe à la fonction l'index dans la figure 2, la fonction crée et remplit le fichier texte "out.txt" dans la fig. 3. Comme on peut le remarquer, les lettres correspondants aux listes chainées vides sont ignorées.

in.txt

Julia was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once she had peeped into the book her sister was reading, but it had no pictures or conversations in it. "And what is the use of a book", thought Julia, "without pictures or conversations?"

fig. 1

index

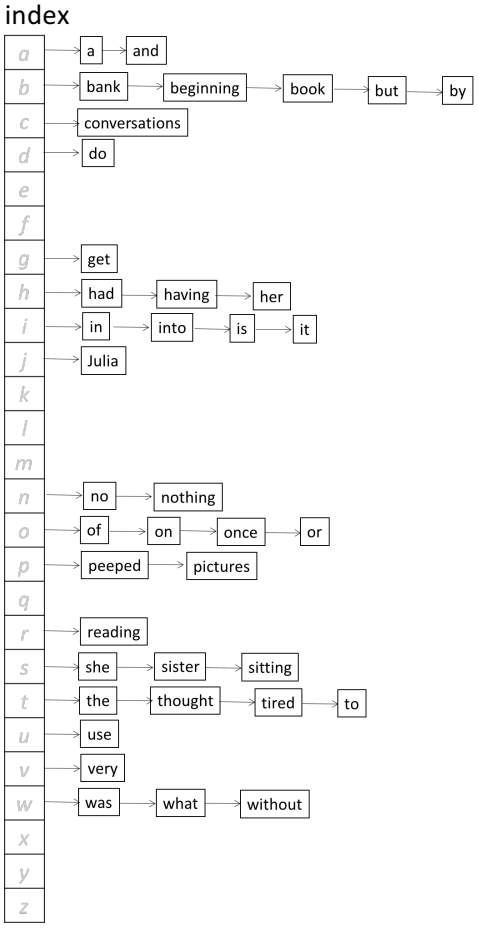

fig. 2

out.txt

a: a - and
b: bank - beginning - book - but - by
c: conversations
d: do
g: get
h: had - having - her
i: in - into - is - it
j: Julia
n: no - nothing
o: of - on - once - or
p: peeped - pictures
r: reading
s: she - sister - sitting
t: the - thought - tired - to
u: use
v: very
w: was - what - without

fig. 3