Licence d'informatique
Module de C/Unix

Examen – Partie C

Jean-Luc LEVAIRE

Septembre 2002

Les documents de cours et TD sont autorisés.
On rendra deux copies séparées pour la partie Unix et pour la partie langage. La partie C est à rendre sur une *copie de couleur blanche.*

1 Questions de cours

Question 1
Répondez le plus précisément possible à ces questions.

1. La déclaration suivante est-elle valide ? `int n = 10; int t[n];`
2. Qu'est-ce qu'un descripteur de fichiers sous Unix ?
3. Quel est le contenu des blocs de données d'un répertoire ? □

Question 2
Que signifient les déclarations suivantes ?

1. `int (*f(int))(int);`
2. `int map(int, int (*f)(int ));` □

2 Gestion de stocks

On désire gérer les stocks d'articles d'une entreprise. Un article est représenté par un numéro `num` (entier), une désignation `des` (chaîne), une quantité `stock` en stock (entier) et un prix de vente `prix` (flottant). L'ensemble de ces articles est stocké (informatiquement) dans une liste chaînée `Art`. D'autre part, on dispose d'une liste de commandes de clients. Une commande est désignée par un numéro (entier), et consiste en une liste de quantités commandées (entier) pour chaque article (`Item`) de la commande. Les commandes sont aussi stockées informatiquement sous forme d'une liste. Les items des commandes désignent leurs articles associés par un pointeur sur la structure `Art` associée. La figure 1 présente un exemple de cette structure de données.

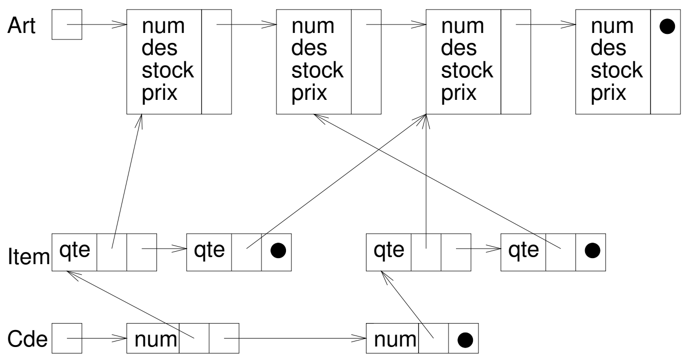

FIG. 1 – Sructure de données de gestion de stocks.

Id: xamc02-sep.tex,v 1.2 2002/09/10 16:21:39 marquet Exp

Question 3
Écrire les types `Art`, `Item`, et `Cde`, ainsi que les types `nodeart`, `nodeitem` et `nodecde` correpondant aux nœuds des différentes listes. ☐

Question 4
Écrire une fonction `int enterstock(int numart, int qte, Art art)` qui modifie la quantité en stock de l'article `numart` dans la liste `art`, de la quantité `qte`. Cette fonction renvoie 0 si l'article est inexistant, et 1 sinon. ☐

Question 5
Écrire une fonction `void printcdes(Art art, Cde cde)` qui affiche pour chaque article de `art`, la somme des quantités commandées par les commandes de la liste de commandes `cde`. ☐

Question 6
Écrire une fonction `int checkstock(int numcde, Cde cde)` qui renvoie 0 si la commande de numéro `numcde`, de la liste de commandes `cde`, peut être satisfaite, et 1 sinon. ☐

Question 7
Écrire une fonction `int docde(int numcde, Cde cde)` qui facture la commande de numéro `numcde` de la liste de commandes `cde`. On supposera que la commande peut être satisfaite. Cette fonction modifie les quantités en stock des articles commandés, supprime la commande de la liste `cde`, et renvoie le montant total à facturer. ☐

Question 8
Écrire une procédure

```
void insertcde(int numcde, struct linecde lines[], int nlines,
               Cde cde, Art art)
```

qui insère une nouvelle commande de numéro `numcde` dans la liste `cde`. Cette insertion se fera en queue de liste (pour conserver l'ordre chronologique des commandes). `lines` est un tableau de `nlines` lignes de commandes, chaque ligne de commande donnant un numéro d'article de la liste d'articles `art` et une quantité commandée :

```
struct linecde { int numart; int qte };
```

☐

Fin du sujet