Auteur : Mohamed Messabihi
Matière : Programmation et structures de données
Date 22 Mai 2016
Durée 1h30

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
1ère Année MI
Semestre 2

Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en **C**
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Trouver les 7 erreurs — 7 pts pts. 20'

Le programme ci-dessous est volontairement truffé d'erreurs (par le méchant Toto), lors de sa compilation le compilateur signale la présence d'au moins 7 erreurs. Identifiez puis corrigez les.

```c
typedef  Etudiant struct Etudiant;
struct Etudiant {
    char nom[30];
    int age;
    float moyenne;
}
void saisirEtudiant(Etudiant *e){
    puts("Donnez le nom : ");
    scanf("%s", & e.nom);
    puts("Donnez l'age : ");
    scanf("%d", &e->age);
    puts("Donnez la moyenne : ");
    scanf("%lf", & *e.moyenne);
}
void main(){
    Etudiant e1, e2={"Toto", 12, 12.65}  ;
    saisirEtudiant(e1);
    e1.nom = "Toto";
    if ( e1.nom == e2.nom)
        printf("l'etudiant e1 est aussi un Toto");
}
```

2 Gestion d'un magasin — 14 pts. 1h10

On souhaite écrire un programme qui permet de gérer un magasin vendant des produits électroménagers . Ce programme permet, entre autres, de saisir les produits, de créer des commandes et d'établir des factures.

Un produit est défini par :

- reference : la référence du produit (par exemple 12985) ;
- designation : la désignation du produit (par exemple Micro-onde) ;
- prixUnitaire : le prix unitaire du produit (par exemple 9850 DA) ;

1. Définir la structure **Produit**. **2 pts**

2. Un catalogue contient le nombre de produits et l'ensemble des produits existants dans le magasin. Proposez une structure pour le Type **Catalogue**. **1 pt**

3. Écrire les deux fonctions **saisir_Produit** et **saisir_Catalogue**. **2 pt**

4. Écrire une fonction **chercher_Reference** qui prend en entrée une référence d'un produit et un catalogue et qui retourne l'indice du produit correspondant dans le catalogue. Si le produit n'existe pas dans le catalogue, la fonction renvoie -1. **2 pts**

5. Une commande contient plusieurs lignes de commande. Chaque ligne de commande contient une référence d'un produit ainsi que sa quantité. Une commande ne doit pas contenir plus de 20 lignes. Définir la structure **Commande**. **2pts**

6. Écrire une fonction **saisir_Commande** qui prend en entrée une commande et un Catalogue et qui demande à l'utilisateur de saisir les références des produits ainsi que leurs quantités. **2pts**

   ☞**Remarque.**
   - La fonction ne doit accepter que les références des produits existants dans le catalogue.
   - La fonction ne doit accepter que des quantités strictement supérieurs à 0.

7. Écrire une fonction **afficher_Facture** qui prend en entrée une commande et un catalogue et affiche la facture à l'écran. Pour chaque référence de produit commandé, la fonction affiche sa référence, sa désignation, son prix unitaire, sa quantité dans la commande et le prix total de la ligne. A la fin de la facture, la fonction affiche le montant total à payer. **2 pts**

   L'affichage de cette fonction doit ressembler celui montré ci-dessous :

```
------------------------------ Facture ----------------------------------
Reference     Designation      Prix Unitaire       Quantite        Total
45665         Téléviseur       66500DA             3             199500DA
12985         Micro-onde       9850DA              2              19700DA
87653         Réfrigérateur    86000DA             1              86000DA
86509         Cafetière        6500DA              4              26000DA
-------------------------------------------------------------------------
                                                  Montant Total = 332200DA
```

8. Écrire une fonction **main** qui permet de saisir un catalogue, créer une commande puis d'afficher sa facture. **1pt**

Bon courage !»

Auteur : Mohamed Messabihi
Matière : Programmation et structures de données
Date 22 Mai 2016
Durée 1h30

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
1ère Année MI
Semestre 2

Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en **C**
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Trouver les 7 erreurs — 7 pts pts. 20'

Le programme ci-dessous est volontairement truffé d'erreurs (par le méchant Toto), lors de sa compilation le compilateur signale la présence d'au moins 7 erreurs. Identifiez puis corrigez les.

Solution

```c
typedef struct Etudiant Etudiant;            // typedef struct Etudiant
struct Etudiant {
    char nom[30];
    int age;
    float moyenne;
};                                           // il manquait le ';'
void saisirEtudiant(Etudiant *e){
    puts("Donnez le nom : ");
    scanf("%s", & e->nom);                   // e-> au lieu de e.
    puts("Donnez l'age : ");
    scanf("%d", &e->age);
    puts("Donnez la moyenne : ");
    scanf("%lf", & (*e).moyenne);            // il manquait les parenthèse (*e)
}
void main(){
    Etudiant e1, e2={"Toto", 12, 12.65};
    saisirEtudiant(&e1);                     // il manquait le '&'
    strcpy(e1.nom , "Toto");                 // strcpy pour copier une chaine
    if ( ! strcmp(e1.nom , e2.nom))          // strcmp pour comparer deux chaines
        printf("l'etudiant e1 est aussi un Toto");
}
```

2 Gestion d'un magasin — 14 pts. 1h10

On souhaite écrire un programme qui permet de gérer un magasin vendant des produits électroménagers . Ce programme permet, entre autres, de saisir les produits, de créer des commandes et d'établir des factures.

Un produit est défini par :

- reference : la référence du produit (par exemple 12985) ;
- designation : la désignation du produit (par exemple Micro-onde) ;
- prixUnitaire : le prix unitaire du produit (par exemple 9850 DA) ;

1. Définir la structure **Produit**. **2 pts**

Solution

```c
typedef struct Produit Produit ;
struct Produit {
    int reference;
    char designation[30];
    double prixUnitaire;
};
```

2. Un catalogue contient le nombre de produits et l'ensemble des produits existants dans le magasin. Proposez une structure pour le Type **Catalogue**. **1 pt**

Solution

```c
typedef struct Catalogue Catalogue;
struct Catalogue {
    int nbrProduits;
    Produit listeProduits[100];
};
```

3. Écrire les deux fonctions **saisir_Produit** et **saisir_Catalogue**. **2 pt**

Solution

```c
void saisir_Produit(Produit *p){
    puts("Donnez la reference du produit : ");
    scanf("%d", &p->reference);
    puts("Donnez la designation : ");
    scanf("%s", &p->designation);
    puts("Donnez le prix unitaire : ");
    scanf("%lf", &p->prixUnitaire);
}
void saisir_Catalogue(Catalogue *c){
    int i=0;
    puts("---- saisir d'un catalogue----");
    puts("Donnez le nombre de produits");
    scanf("%d", & c->nbrProduits);
    for (i=0; i<c->nbrProduits; i++)
    {
        saisir_Produit(& c->listeProduits[i]);
    }
}
```

4. Écrire une fonction **chercher_Reference** qui prend en entrée une référence d'un produit et un catalogue et qui retourne l'indice du produit correspondant dans le catalogue. Si le produit n'existe pas dans le catalogue, la fonction renvoie -1. **2 pts**

Solution

```c
int chercher_reference(int  r, Catalogue c){
  int i=0;
  while (c.listeProduits[i].reference !=r && i < c.nbrProduits) i++;
  if (i < c.nbrProduits) return i;
  else return -1;
}
```

5. Une commande contient plusieurs lignes de commande. Chaque ligne de commande contient une référence d'un produit ainsi que sa quantité. Une commande ne doit pas contenir plus de 20 lignes. Définir la structure **Commande**. **2pts**

Solution

```c
typedef struct LigneCommande LigneCommande;
struct LigneCommande {
    int reference;
    int quantite;
};
typedef struct Commande Commande;
struct Commande {
    int numero;
    int nbrProduits;
    LigneCommande lignes[20];
};
```

6. Écrire une fonction **saisir_Commande** qui prend en entrée une commande et un Catalogue et qui demande à l'utilisateur de saisir les références des produits ainsi que leurs quantités. **2pts**

   ☞**Remarque.**
   - La fonction ne doit accepter que les références des produits existants dans le catalogue.
   - La fonction ne doit accepter que des quantités strictement supérieurs à 0.

Solution

```c
void saisir_LigneCommande(LigneCommande *lc, Catalogue c){
    do{
        puts("Donnez la reference du produit : ");
        scanf("%d", &lc->reference);
    } while( chercher_reference(lc->reference, c) == -1);
    do {
        puts("Donnez la quantite (positive)");
        scanf("%d", &lc->quantite);
    }while (lc->quantite <=0);
}
void saisir_Commande(Commande *c, Catalogue cat){
    int i=0;
    puts("Donnez le nombre d'articles a commander ");
    scanf("%d", &c->nbrProduits);
    for (i=0; i<c->nbrProduits; i++)
    {
        saisir_LigneCommande(& c->lignes[i], cat );
    }
}
```

7. Écrire une fonction **afficher_Facture** qui prend en entrée une commande et un catalogue et affiche la facture à l'écran. Pour chaque référence de produit commandé, la fonction affiche sa référence, sa désignation, son prix unitaire, sa quantité dans la commande et le prix total de la ligne. A la fin de la facture, la fonction affiche le montant total à payer. **2 pts**

   L'affichage de cette fonction doit ressembler celui montré ci-dessous :

```
------------------------------ Facture ---------------------------------
Reference     Designation      Prix Unitaire       Quantite        Total
45665         Téléviseur       66500DA             3             199500DA
12985         Micro-onde       9850DA              2              19700DA
87653         Réfrigérateur    86000DA             1              86000DA
86509         Cafetière        6500DA              4              26000DA
------------------------------------------------------------------------
                                                Montant Total = 332200DA
```

Solution

```
1 void afficher_Facture (Commande c, Catalogue cat){
2     int i,k;
3     double montantPayer = 0;
4     Produit p;
5     puts("-------------------------- Facture ----------------------------");
6     puts("Reference     Designation     PrixUnitaire    quantite    Total");
7     for (i=0; i<c.nbrProduits; i++)
8     {
9         k = chercher_reference(c.lignes[i].reference, cat);
10        p = cat.listeProduits[k];
11        printf("%d       %s              %f                %d            %f\n
              ", p.reference, p.designation, p.prixUnitaire,  c.lignes[i].
              quantite, c.lignes[i].quantite*p.prixUnitaire );
12        montantPayer = montantPayer + c.lignes[i].quantite*p.prixUnitaire;
13    }
14    puts("------------------------------------------------------------------");
15    printf("                              Montant Total =  %f", montantPayer);
16 }
```

8. Écrire une fonction **main** qui permet de saisir un catalogue, créer une commande puis d'afficher sa facture. **1pt**

Solution

```
1 void main(){
2     Catalogue Cat;
3     Commande Com;
4     saisir_Catalogue(&Cat);
5     saisir_Commande(&Com, Cat);
6     afficher_Facture(Com, Cat);
7 }
```

Bon courage !»