esprit Se former autrement	Semestre : 1 ■ 2 □ Session : Principale □ Rattrapage ■

Module : Programmation procédurale 1

Enseignants : Équipe Programmation procédurale 1

Classes : 1A1 -> 1A30

Documents autorisés : OUI ■ NON □ Nombre de pages : 04

Date : 25/06/2021 Heure : 13H00 Durée : 1H30

Connexion autorisée : OUI □ NON ■

N.B. *:Vous devez utiliser la programmation modulaire (.h et .c)*

Enoncé :

Une société de vente en ligne vous a confié le développement d'une application permettant de gérer les livreurs de ses commandes en ligne.

1. Un livreur est caractérisé par :
   - Un identifiant **unique** : chaîne de caractères
   - Un nombre de commande : entier
   - Un prix total des commandes : réel
   - Un tableau de **commande**s
2. Une commande est caractérisée par
   - Un code **unique** : entier
   - Un prix : réel
   - Un état : chaîne de caractères (confirmée ou non confirmée)

Pour cela, on a besoin d'enregistrer les différents livreurs dans **un tableau.**

A. Définir les structures nécessaires en utilisant ***typedef***.

B. Développez alors :

La fonction principale **int main()** contenant un menu qui assure le bon fonctionnement de l'application et qui assure les fonctionnalités suivantes :

- Ajouter un nouveau livreur
- Afficher les livreurs
- Ajouter une commande à un livreur
- Confirmer une commande
- Afficher le nombre de commandes non confirmées d'un livreur donné
- Afficher le prix total des commandes et le pourcentage des commandes non confirmées de tous les livreurs
- Supprimer une commande

La fonction « main » doit être mise à jour au fur et à mesure du développement des différentes fonctions qui suivent :

1. **void saisir_livreur (Livreur *l)** qui permet de saisir les informations d'un livreur. A l'état initial, le livreur **n'a aucune commande** et **le prix total des commandes est nul**. [Voir la figure 1].

2. **int chercher_livreur (Livreur tabL[], int nbL, char idL[])** qui permet de chercher un livreur donné par son identifiant et de retourner sa position dans le tableau des livreurs **tabL** s'il existe et -1 sinon.

3. **void ajouter_livreur (Livreur tabL[], int* nbL, Livreur l)** qui permet d'ajouter un livreur **l** au tableau des livreurs **tabL.** La fonction doit vérifier si l'identifiant du livreur à ajouter existe ,si c'est le cas un message d'erreur sera affiché.

   Exemple :

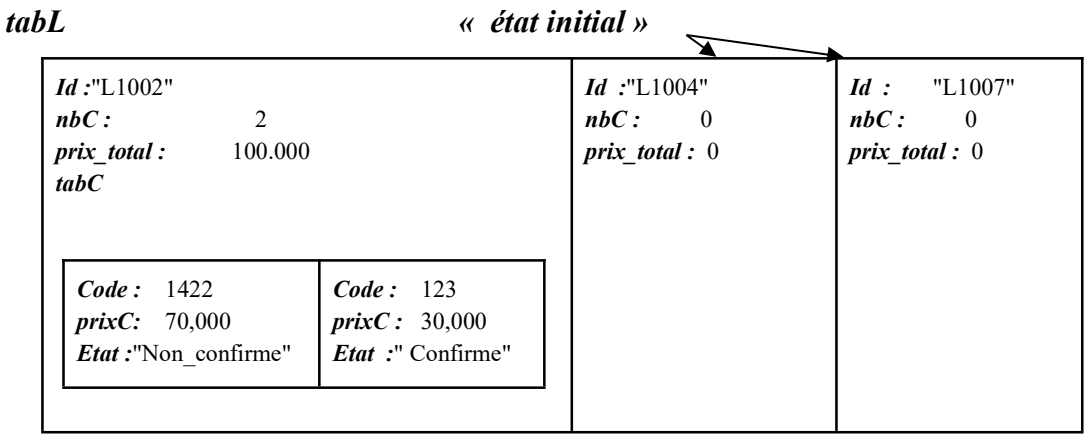

Figure 1 : Exemple d'un tableau de livreurs

4. **void afficher_livreur (Livreur tabL[], int nbL)** qui permet d'afficher pour chaque livreur son identifiant, le nombre de ses commandes, le prix total des commandes , le code de chaque commande ainsi que son état.
5. **Commande saisir_commande()** qui permet de saisir les informations d'une commande tout en initialisant son état à la chaîne « **non_confirmé** ».
6. **int chercher_commande (Livreur tabL[], int nbL, char idL[], int code)** qui permet de chercher une commande donnée par son **code** d'un livreur dont l'identifiant **idL** est passé en paramètre. La fonction retourne la position de la commande si elle existe, -1 sinon.
7. **void ajouter_commande (Livreur tabL[], int nbL, char idL [], Commande c)** qui permet d'ajouter une commande **c** à un livreur donné par son identifiant **idL** tout en vérifiant l'unicité de la commande et de mettre à jour le nombre des commandes de ce livreur**.**
8. **void confirmer_commande (Livreur tabL[], int nbL, char idL [], int code)** qui permet de confirmer une commande dont le code est passé en paramètre d'un livreur donnée par son identifiant **idL** s'il existe**,** sinon un message d'erreur sera affiché.
   La confirmation d'une commande consiste à mettre à jour l'état de la commande et le prix total des commandes.
9. **int non_confirmee (Livreur tabL[], int nbL, char idL [])** permettant de retourner le nombre total des commandes non confirmées d'un livreur donnée par son identifiant **idL**.
10. **void poucentage_nonconf (Livreur tabL [], int nbL, float * totComd, float *prc)** qui permet de retourner le prix total des commandes **totCmd** et le pourcentage des commandes non confirmées **prc** de tous les livreurs.
11. **void supprimer_commande (Livreur tabL[], int nbL, char idL [], int code)** qui permet de supprimer une commande donnée par son **code** d'un livreur dont l'identifiant **idL** est passé en paramètre. un message d'erreur sera affiché si la commande ou le livreur n'existe pas.

Bon Travail

Traitement	Algorithmique
Structure	**1**
Modularité	**0.5**
Saisir un livreur	**0.75**
Saisir une commande	**1**
Chercher un livreur	**1**
Chercher une commande	**1.5**
Ajouter un nouveau livreur	**1.5**
Ajouter une commande à un livreur donné par son identifiant	**1.5**
Afficher les livreurs	**1**
Nombre de commandes non confirmées	**1.5**
Confirmer une commande	**1.5**
Afficher le pourcentage et le prix total des commandes	**2**
Supprimer une commande	**1.25**
Total	**16**

Traitement	Exécution	/ Appel /	Total
Menu	0.25	0.25	**0.5**
Ajouter un nouveau livreur	0.25	0.25	**0.5**
Ajouter une commande	0.25	0.25	**0.5**
Afficher les livreurs	0.25	0.25	**0.5**
Afficher le nombre des commandes non confirmées d'un livreur donné	0.25	0.25	**0.5**
Confirmer une commande	0.25	0.25	**0.5**
Afficher le prix total des commandes et le pourcentage des commandes non confirmées de tous les livreurs	0.25	0.25	**0.5**
Supprimer une commande	0.25	0.25	**0.5**
Total	2	2	**4**