

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

Module : Programmation Procédurale I

Enseignant(s) : Équipe Programmation C

Classe(s): 1A1, 1A2, 1A3, 1A4, 1A5

Documents autorisés : OUI ☒ NON ☐ Nombre de pages : 2

Connexion autorisée : OUI ☐ NON ☒

Date : 06/01/2016

Heure :9h

Durée :1h.30

Vous devriez utiliser les bonnes pratiques de développement :

La programmation modulaire (.h / .c)

Des noms de variables et de fonctions significatifs

### Énoncé

Une pâtisserie vous a confié le développement d'une application pour la gestion des ventes de gâteaux.

Un **gâteau** est caractérisé par:

Un type (**0**:chocolat, **1**:nougatine, **2**:pistache)

Un nombre d'ingrédients

Un tableau d'ingrédients

Un **ingrédient** est caractérisé par :

Une désignation (chaîne de caractères)

Une quantité ( entier)

Un prix unitaire ( réel)

Développez alors:

Une fonction principale **main** contenant un menu qui assure le bon fonctionnement de l'application. La fonction main doit être mise à jour au fur et à mesure du développement des différentes fonctions qui suivent:

1. **void saisir (gâteau \* g)** qui permet de saisir les informations d'un gâteau.
2. **void ajouterGâteau ( gâteau tabG [], gâteau g, int \* tg)** qui permet d'ajouter un gâteau **g** dans le tableau **tabG** et de mettre à jour sa taille **tg**.
3. **void afficher(gâteau tabG [], int tg)** qui permet d'afficher le tableau des gâteaux **tabG** .
4. **void trouver(gâteau tabG[],int tg, int type, int \*pos)** qui permet de renvoyer la position **pos** dans le tableau **tabG** du premier gâteau rencontré dont le type est passé en paramètre.

5. **float calculerPrix(gateau g )** qui permet de retourner le prix du gâteau passé en paramètre selon son type. Sachant que le  $\text{Prix} = \text{coût} + \text{gain}$ , avec :

$$\text{coût} = \sum_{i=0}^{\text{nombre d'ingrédient}} (\text{prix unitaire}_{\text{Ingrédient } i} * \text{quantite}_{\text{Ingrédient } i}).$$

Le gain est de 20% du coût pour le type pistache, 15% du coût pour le type nougatine et 10% du coût pour le type chocolat.

On voudrait gérer les ventes de la pâtisserie par semaine. Pour ce la, on dispose d'une matrice, contenant le nombre des ventes par jour pour chaque type de gâteau. A chaque fois qu'un client se présente pour acheter un gâteau, on incrémente la valeur de la case qui correspond au type désiré et au jour de vente dans la matrice **matV** (ci-dessous un exemple).

	lundi	mardi	mercredi	jeudi	vendredi	samedi	dimanche
chocolat	20	33	0	1	2	1	1
nougatine	15	100	1	2	3	2	2
pistache	12	55	5	2	3	2	3

**Exemple :** Le nombre de gâteaux chocolat vendus le lundi est **20**.

On dispose aussi d'un tableau de 7 cases **tabP** pour stocker le chiffre d'affaire<sup>1</sup> réalisé pour chaque jour de la semaine.

Pour ce faire, implémentez les trois fonctions suivantes:

6. **void initialiser(int matV [][7], float tabP[])** qui permet d'initialiser la matrice de vente **matV** et le tableau des prix **tabP** à 0.
7. **void vendre(int matV [][7], gâteau tabG[], int type, int jour, int \*tg, float tabP[])** qui permet de vendre le premier gâteau stocké dans le tableau **tabG** selon le type passé en paramètre en procédant comme suit:
  - calculer son prix de vente et mettre à jour le tableau de chiffre d'affaire **tabP**.
  - mettre à jour la matrice **tabV** en incrémentant la valeur de la case correspondante.
  - supprimer le gâteau du tableau **tabG**.
8. **void afficherRecette(int matV [][7], int jour, float tabP[])** qui permet d'afficher pour un jour donné, le chiffre d'affaire et la quantité de gâteaux vendue pour chaque type.

**Bonne Chance**

<sup>1</sup> le chiffre d'affaire correspond au prix totale de vente.