

DS N° 1 INFORMATIQUE II

Calculatrice et documents non autorisés
Les réponses pour la partie code doivent être données en C**Exercice 1 (Gestion d'une base de données clientèle) (13.5 pts)**

> Si vous ne parvenez pas à répondre à une question, vous pouvez passer à la suivante en supposant que la réponse précédente a été fournie correctement).

> Les fonctions déjà écrites peuvent être ré-utilisées dans les questions suivantes.

> On ne vous demande pas d'écrire la fonction principale du programme.

Un commerce souhaite gérer automatiquement sa base de clients, leurs données personnelles ainsi que les réductions liées à leur fidélité/ancienneté.

1. Un `client` est défini par :

- son nom,
- son prénom,
- son adresse,
- ses points de fidélité (positif ou nul),
- sa date de naissance,
- sa date d'adhésion au programme de fidélité du commerce.

avec une `adresse` qui est définie par :

- un numéro de rue (strictement positif),
- un nom de rue,
- un nom de ville.

et une `date` définie par :

- un numéro de jour (entre 1 et 31 inclus),
- un numéro de mois (entre 1 et 12 inclus),
- un numéro d'année (positif ou nul).

Définir la ou les structures nécessaire(s) pour gérer la base de clients du magasin.

2. Écrire une fonction/procédure `afficheAdresse(...)` permettant d'afficher les informations d'une `adresse` passée en paramètre.
3. Écrire une fonction/procédure `memeDate(...)` qui prendra deux `dates` en paramètre et va retourner 1 si les deux date représente le même jour de l'année, 0 sinon. Ainsi, **deux dates sont considérées identiques si le jour et le mois sont équivalents**.
4. Écrire une fonction/procédure `nombreAnnees(...)` qui va prendre deux `date` en paramètre et qui va retourner le nombre d'années entières et révolues qui les sépare (il ne faut donc pas simplement récupérer la différence des années).
5. Écrire une fonction/procédure `memeMot(...)` qui prendra en paramètre deux chaînes de caractères et qui retournera 1 si elles sont les mêmes, 0 sinon.
6. Écrire la fonction constructeur d'un `client` permettant à l'utilisateur de saisir les données personnelles du client. Les points de fidélité démarrent à 0 et la date d'inscription sera celle d'aujourd'hui (nous sommes le 23/02/2023). Vous devez tester que les données entrées par l'utilisateur sont correctes, et sinon lui demander de recommencer. On ne vérifiera pas la cohérence entre le nombre de jours et le mois, on vérifiera simplement les plages de valeurs.
7. Les points de fidélité d'un client augmentent de 1 pour chaque tranche d'achats de 100 Euros. Il n'y a que des points entiers.
Écrire une fonction/procédure `majPoint(...)` qui prendra en paramètre un `client` ainsi que le montant de ses achats, et qui retournera ses points de fidélité mis à jour.
8. Un client bénéficie d'une réduction qui est calculée comme suit :

- Chaque année d'inscription au programme fidélité jusqu'à aujourd'hui (nous sommes en 2023) ajoute 5% de réduction.
 - Chaque point de fidélité ajoute 1% de réduction.
 - Le taux de réduction est plafonné à 50%.
- Écrire une fonction/procédure `reduction(...)` qui prendra en paramètre un `client` et qui retournera le taux de réduction de son prochain achat.

Nous allons maintenant écrire des fonctions permettant de gérer l'ensemble des clients du commerce.

9. Le commerce tient à être chaleureux avec ses fidèles clients et a pour tradition de leur fêter leur anniversaire. Écrire une fonction/procédure `anniversaire(...)` qui prendra en paramètre :
- un tableau de `client`
 - la taille de ce tableau
 - une `date`
- Elle devra afficher le nom et prénom des clients du tableau dont l'anniversaire est à la date donnée en paramètre.
10. Le commerce souhaite effectuer des campagnes de publicité ciblées géographiquement. Écrire une fonction/procédure `adressePublicite(...)` qui prendra en paramètre :
- un tableau de `client`
 - sa taille
 - une chaîne de caractères qui stockera le nom d'une ville.
- Ce code devra afficher l'adresse des clients dont la ville de résidence est la ville donnée en paramètre.

Exercice 2 (Tableau trié ?) (2.5 pts)

- Écrire une fonction `tableauCroissant(...)` qui :
- prendra en paramètre un tableau d'entiers et sa taille,
(chaque entier de ce tableau sera strictement positif et avec une valeur bornée entre 0 et 255 incluses)
 - vérifiera que ce tableau est trié dans l'ordre croissant (i.e. l'élément de chaque case est supérieur ou égal à l'élément de la case précédente),
 - utilisera le formalisme pointeur pour parcourir le tableau.

Exercice 3 (Tri de 3 valeurs) (4.0 pts)

Écrire un programme qui demande à l'utilisateur de saisir 3 valeurs réelles `a`, `b`, `c`, qui seront passées ensuite en paramètres à une fonction/procédure `traitement(...)`. A l'issue de l'exécution de `traitement(...)`, on veut que la plus petite valeur soit stockée dans `a`, la plus grande valeur soit stockée dans `c`, et la valeur intermédiaire stockée dans `b`.