



## Examen

<b>Module:</b> Programmation Procédurale	<b>Classes:</b> 1 <sup>ères</sup> Années Minds, Info, GI, Tél, GE
<b>Documents:</b> Non autorisés	<b>Durée:</b> 1h30
<b>Professeurs :</b> Zoghalmi M.A., Alouane S., Sbaï Z., Hamrouni K. Frihida A.	<b>Date :</b> 23 Janvier 2014

### Exercice : (6 pts)

- 1- On voudrait implémenter une pile d'au maximum 1000 caractères.
  - a. Définir la structure de données « PILE ».
  - b. Ecrire les fonctions suivantes :
    - **void InitPile (PILE \*first)** : permettant d'initialiser la pile pointée par **first**.
    - **int Empiler (PILE \* first, char c)** : permettant d'empiler le caractère **c** sur la pile pointée **first**. Si la pile est pleine, la fonction **Empiler** renvoie 0 sinon elle renvoie 1.
    - **int Depiler (PILE \* first, char \*c)** : permettant de dépiler un caractère et de le ranger dans **\*c**. Si la Pile est vide, la fonction **Depiler** renvoie la valeur 0, sinon elle renvoie la valeur 1.
- 2- Écrire une fonction **int Parentheses(char \*s)** prenant en paramètre une chaîne de caractères et vérifiant que cette chaîne est bien parenthésée. Exemple : la chaîne "2(j(2/3 \* 2))" est bien parenthésée alors que les chaînes "(2+3)\*5 + (4 \* (3+2))" et "2((mot \* 5) +4 \* (i-))" ne sont pas bien parenthésées.

### Problème ( 14 pts):

On se propose de créer un programme en C permettant de manipuler un ensemble de comptes bancaires stockés dans un tableau **Tab** ayant une longueur maximale égale à 1000. Chaque compte bancaire peut subir des opérations de dépôt/ajout ou bien de retrait. Toutes les opérations effectuées sur un compte sont enregistrées sous la forme d'une liste chaînée. Soient les définitions suivantes :

Une « **DATE** » est composée de 3 entiers **jj**, **mm**, **aa** donnant respectivement le jour, le mois et l'année.

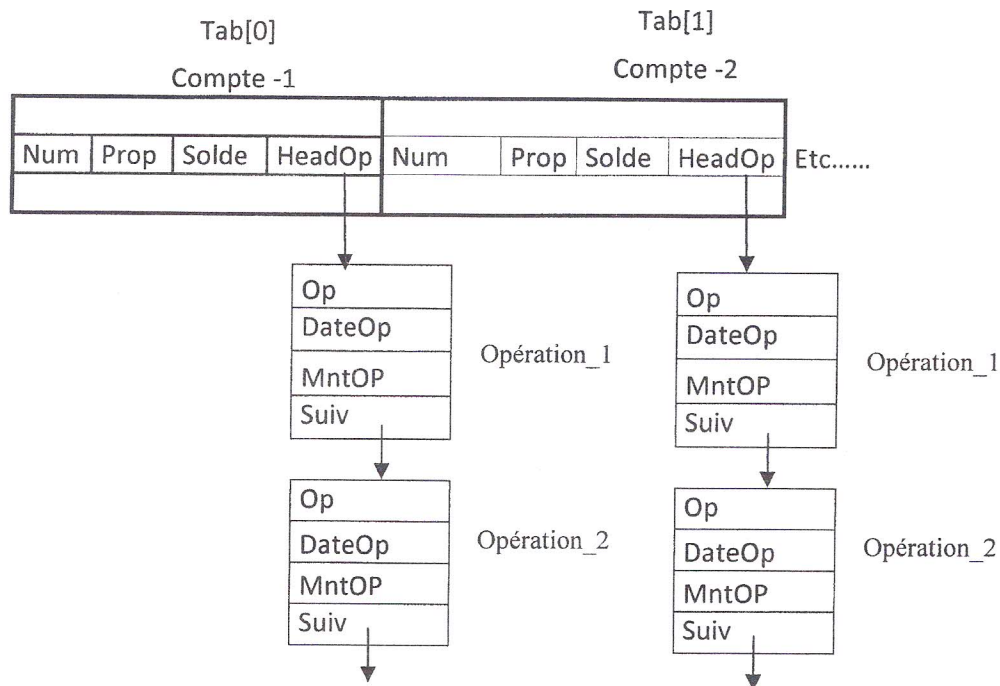
Une « **OPERATION** » est composée de :

- **Op** : (type caractère) qui peut prendre l'une des valeurs suivantes : « C » s'il s'agit d'un ajout au solde, ou « D » s'il s'agit d'un retrait du solde.
- **DateOp** : (type DATE) donnant la date de l'opération.
- **Montant** : (type réel) donnant le montant de l'opération d'ajout ou de retrait.
- **Suiv** : un pointeur sur l'opération suivante.

Un « **COMPTE** » est composé de :

- **Num** : (entier) donnant le numéro (unique) d'identification du compte
- **Prop** : (chaîne de caractères, 20 au max) donnant le nom du propriétaire du compte
- **Solde** : (type réel) : le solde du compte
- **HeadOp** : un pointeur de type **OPERATION** donnant le début de la liste chaînée des opérations « type **OPERATION** » effectuées sur le compte.

- 1- Définir les types **DATE**, **OPERATION** et **COMPTE**, selon le schéma ci-après :



2- Ecrire les fonctions suivantes :

- int ChargerCompte (COMPTE Tab [ ], int TailleMax):** Cette fonction prend comme paramètre le tableau **Tab** de taille maximale **TailleMax**, initialement vide, elle permet d'accepter, à partir du clavier, tous les comptes et de les ranger dans le tableau **Tab**. La fin de la saisie est détectée quand l'utilisateur tape -1 comme numéro de compte. Cette fonction retourne le nombre exact de comptes saisis. **Attention** : cette fonction ne saisit pas d'opérations.
- float Ajouter (COMPTE Tab [ ], int N, float montant, int num, DATE d):** Cette fonction prend comme paramètre le tableau **Tab** de **N** comptes, permet de déposer, à la date **d**, le **montant** d'argent dans le compte numéro **num**. L'opération d'ajout sera insérée à la tête de la liste d'opérations. Cette fonction retourne le nouveau solde du compte;
- int Rechercher (COMPTE Tab [ ], Int N, int num):** permet de chercher le compte numéro **num** dans le tableau **Tab** de **N** comptes. Cette fonction retourne 1 si le compte existe dans le tableau, 0 sinon.
- int AnnulerOperation (COMPTE Tab [ ], Int N, int num):** permet d'annuler la dernière opération effectuée sur le compte de numéro **num**.
- int Supprimer (COMPTE Tab [ ], int N, int num):** Cette fonction prend comme paramètre le tableau **Tab** de **N** comptes, permet de supprimer le compte numéro **num** du tableau **Tab**. Elle retourne le nouveau nombre de comptes.
- void Consulter (COMPTE Tab [ ], int N, int num):** Cette fonction prend comme paramètre le tableau **Tab** de **N** comptes, permet de consulter le compte numéro **num**. Elle affiche le numéro du compte, le nom du propriétaire, et le solde actuel.

#### Barème

Exercice		Problème						
1	2	1	2.a	2.b	2.c	2.d	2.e	2.f
3	3	1.5	2	2.5	2	2	2	2

*Bon travail.....*