

Devoir surveillé <input type="checkbox"/>	Examen <input checked="" type="checkbox"/>	Session : principale <input checked="" type="checkbox"/> de contrôle <input type="checkbox"/>
Matière : Algorithmique et structures de données I	Semestre: 1	
Enseignant : Aymen Sellaouti et Wided Miled	Date: 05/01/2016 à 15h00	
Filière(s) : MPI	Durée: 1h30	
Nombre de pages : 2	Documents : autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/>	

### Exercice 1 (6 pts) :

- Ecrire en **algorithmique** la fonction `est_premier` qui permet de vérifier si un entier est premier ou non. Elle retournera vrai si c'est le cas, faux sinon (2 pts).
- Ecrire en **algorithmique** la fonction `superPremier` qui permet de vérifier si un entier est super premier. Un entier est dit super premier s'il est premier et si tous les entiers qui le composent en enlevant à chaque fois le chiffre de gauche sont aussi premiers (4 pts).

**Exemple** : 523 est super premier vu que 523 est premier, 23 est premier et 3 est premier.

### Exercice 2 (14 pts)

Une agence bancaire utilise pour le suivi des mouvements des comptes de ses clients les deux fichiers texte suivants :

#### « **Compte.txt** »

La structure étant la suivante :

Num_compte	: 6 caractères alphanumériques
Titulaire	: 15 caractères alphabétiques
Adresse	: 15 caractères alphanumériques
Ville	: 10 caractères alphabétiques
Code postal	: 4 caractères numériques
Premier_dans_mvt	: 4 caractères numériques désignant le numéro de la ligne dans le fichier <code>mvt.txt</code> correspondant à la première opération du compte (si le compte n'a aucune opération, <code>Premier_dans_mvt</code> est égal à -1)

**Remarque** : Un numéro de compte dans le fichier « `compte.txt` » est unique.

« **mvt.txt** » : représentant les mouvements bancaires (débit et crédit) de tous les comptes (**trié par ordre croissant de la date d'opération : la dernière ligne du fichier représente le mouvement le plus récent**). La structure étant la suivante :

Num_compte	: 6 caractères alphanumériques
Date_operation	: 10 caractères en format <code>jj/mm/aaaa</code>
Montant	: 10 caractères (format 10.3)
Nature_operation	: un caractère (D : pour débit C : pour crédit)
Désignation_operation	: 10 caractères alphanumériques
Mvt_Suivant	: 4 caractères numériques désignant le numéro de la ligne dans ce fichier correspondant à l'opération suivante pour le même compte (s'il n'y a pas d'opération suivante <code>Mvt_suivant</code> sera égal à -1)

Chaque ligne de ce fichier décrit une opération effectuée sur un compte.

Si la nature de l'opération est 'D' le montant est comptabilisé en moins, si elle est 'C' le montant est comptabilisé en plus.

On vous demande d'écrire les fonctions suivantes :

1) (2 pts)

```
int compare_date(char *date1, char *date2){ }
```

qui permet de comparer deux dates et retourner :

0 si égalité entre les deux dates

une valeur < 0 si date1 est inférieure à date2

une valeur > 0 si date1 est supérieure à date2

**Remarque** : date1 et date sont des chaînes de caractères en format **jj/mm/aaaa**

2) (2 pts)

```
int nb_ligne_dans_mvt (FILE *fp_mvt){ }
```

qui retourne le nombre de lignes dans le fichier « mvt.txt » **sans le parcourir.**

3) (3 pts)

```
int seek_compte (FILE *fp_compte, int num_compte, char *ligne_compte){ }
```

qui permet de pointer, dans le fichier « compte.txt », sur la ligne associée à un numéro de compte et de récupérer la ligne du fichier dans la chaîne ligne\_compte. Si le compte n'existe pas, la fonction retourne 0, sinon elle retourne 1.

4) (3 pts)

```
int seek_premier_mvt (FILE *fp_compte, FILE *fp_mvt, int num_compte,  
char *ligne_mvt){ }
```

qui permet de pointer dans le fichier « mvt.txt » sur la première opération d'un compte donné et de récupérer la ligne du fichier dans la chaîne ligne\_mvt. Si le compte n'a pas d'opération, la fonction retourne 0, sinon elle retourne 1.

**Ps : Il est interdit de parcourir tout le fichier « mvt.txt » pour récupérer le premier mouvement il faut directement pointer sur le mouvement.**

5) (4 pts)

Ecrire la fonction **réursive**

```
Void extrait(FILE *fp_mvt, char *num_compte, int num_ligne){ }
```

qui permet d'afficher l'extrait d'un compte contenant les données du compte, et la liste de ses mouvements (date de l'opération, sa désignation et son montant) **du plus récent au plus ancien.**

Lors de l'appel de la fonction **extrait**, la variable **num\_ligne** représente **le numéro de la ligne de la première apparition du mouvement de ce compte dans le fichier «mvt .txt».**

L'affichage sera comme suit :

COMPTE : <compte> Titulaire : <titulaire> Adresse : <adresse>

Code postal : <code\_postal> Ville : <ville>

<date\_operation> <désignation\_opération> <Montant>

.

.

.

**Remarque** : Pour l'affichage de l'entête contenant les informations du compte on suppose que nous disposons de la fonction `afficher_compte(char *ligne_compte)`.

**Bon Travail**