

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    Semestre : 1 Enseignant : Aymen Sellaouti et Majdi Jribi    Date : 03/01/2019 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)

Dans cet exercice, le mot expression désigne une chaîne de caractères ne contenant que des parenthèses ouvrantes et fermantes comme par exemple "(())", "(()" et "(()))".

Une expression est dite bien parenthésée si le nombre de parenthèses ouvrantes est égal au nombre de parenthèses fermantes, et que les parenthèses sont dans le bon ordre.

Par exemple,

- \_ "(())" est une expression bien parenthésée ;
- \_ "(()" est mal parenthésée

Dans cet exercice vous n'avez le droit qu'à un seul parcours de votre chaîne.

1- Ecrire en **C** une fonction nommée **bien\_parenthesee** qui retourne la valeur 1 si la chaîne passée en paramètre est bien parenthésée, et la valeur 0 dans le cas contraire.

2- Dans cette question on ne travaille qu'avec des expressions bien parenthésées. On appelle factorisation d'une expression bien parenthésée un découpage de cette expression en expressions bien parenthésées consécutives, chacune de ces expressions étant appelée facteur.

Voici quelques factorisations d'expressions bien parenthésées

- \_ "(())" = "(())" (un seul facteur) ;
- \_ "O(())" = "O"."(())" (deux facteurs) ;
- \_ "(())O(())" = "(())"."O"."(())" (trois facteurs).

Ecrire en **C** une fonction nommée **nbre\_facteurs** qui calcule le nombre de facteurs que possède une expression bien parenthésée.

### Exercice 2 (4 pts)

Ecrire en **algorithmique** une fonction ou une procédure qui permet de fusionner deux tableaux d'entiers triés dans l'ordre croissant dans un troisième tableau. Le tableau résultant est aussi trié dans l'ordre croissant. Vous n'avez le droit qu'à un seul parcours de vos deux tableaux. Le troisième tableau ne devra pas contenir de redondance.

### Exercice 3 (10 pts)

Afin de gérer le service maternité d'un hôpital, nous disposons des fichiers texte suivants :

« **bebe.txt** »

qui contient l'ensemble des bébés nés dans le service

Chaque ligne de ce fichier contient les données suivantes :

- Code bébé : 5 caractères (entier)
- Nom bébé : 10 caractères
- Prénom bébé : 10 caractères
- Date naissance : 8 caractères au format suivant aaaammjj (exemple 20180702)
- Date sortie : 8 caractères au format aaaammjj (si le bébé n'est pas encore sortie, la date sera la suivante 00000000)
- Numéro chambre : 3 caractères (entier)

#### « **couveuse.txt** »

Qui contient toutes les informations sur les bébés ayant besoins de couveuse pour terminer leur croissance. Chaque ligne de ce fichier contient les données suivantes :

Code bébé : 5 caractères (entier)  
Date entrée : 8 caractères au format suivant aaaammjj (exemple 20180702)  
Date sortie : 8 caractères au format aaaammjj (si le bébé n'est pas encore sorti, la date sera la suivante 00000000)  
CoutJour : 5 caractères (entier) qui représente le prix moyen d'une journée

#### « **visitesMedecins.txt** »

Qui contient toutes les informations sur les visites d'un médecin à un bébé :

NuméroChambre : 5 caractères (entier)  
Code Medecin : 5 caractères (entier)  
Date Visite : 8 caractères au format suivant aaaammjj (exemple 20180702)

On suppose que vous disposez d'une fonction **getNbJours**(long dateDebut, long dateFin) qui récupère deux dates (au format aaaammjj) et retourne le nombre de jours entre ces deux dates.

Vous disposez aussi d'une fonction **inInterval**(long dateDebut, long dateFin, long date) qui retourne 1 si date est dans l'intervalle [dateDebut, dateFin], 0 sinon.

Ecrire en langage C les fonctions suivantes :

- 1- **int nbJourBebe** (FILE\* fp, int codeBb){ } qui retourne le nombre de jour passé par le bébé dans sa chambre. Si le bébé n'est pas encore sorti, vous affichez le message « le bébé est encore dans la chambre » et vous retournez -1.
- 2- Lors de la sortie du bébé, une facture est générée et la date de sortie est mise à jour dans le fichier bebe.txt.

Ecrire les fonctions suivantes :

2.1 void **updateSortie**(FILE\* fp, long dateSortie, int codeBebe) qui modifie la date de sortie du bébé dans le fichier « bebe.txt » de code **codeBebe** de 00000000 vers **dateSortie**.

2.2 int **getPrixCouveuse**(codeBebe) qui retourne le montant à payer pour la couveuse pour le bébé de code codeBebe.

- 3- Vous disposez de la structure suivante :

```
Typedef struct {  
    int code ;  
    int nbVisite ;  
} Medcin;
```

En utilisant cette structure, écrire la fonction

int **getStatsMedcins** (FILE\* fp, Medcin \*t, long dateDeb, long dateFin)

qui permet à partir du fichier « **visitesMedecins.txt** » de récupérer dans le tableau t le nombre de visite effectué par chaque médecin dans l'intervalle [**dateDebut**, **dateFin**]. Un médecin ne doit apparaître qu'une **seule fois** dans le tableau. La fonction devra retourner le nombre de médecin ayant effectué une visite dans cet intervalle.

Bonne chance ☺