

D'APRES DS DU 3 MARS 2007

Il est recommandé de:

- sauvegarder régulièrement son travail,
- tester, dans la mesure du possible, ses fonctions au fur et à mesure de leur implémentation,
- respecter les règles de codage,
- éliminer les warning lors de la compilation.
- penser à fermer les fichiers ouverts et à libérer la mémoire si elle n'est pas utilisée.

Programme à réaliser sous linux et de sauvegarder son programme de la façon suivantes : <1^{er} lettre du prénom><nom>_Test_20070303.c

(exemples : svera_Test_20070303.c, cpaul_Test_20070303.c)

récupérer

Présentation du sujet

Le ping (Parti des Informaticiens Non Gâteaux) souhaite recueillir ses 500 signatures auprès des maires de communes. Il s'agit d'utiliser un arbre binaire pour la gestion de ces données. L'arbre binaire est composé de nœuds de type :

```
typedef struct tnoeud {  
    char nomCommune[32];  
    char nomMaire[32];  
    char codePostal[6];  
    char signature[4]; /* "non" ne donne pas, "???" pas consulté, "oui" ok */  
    struct tnoeud* pFg;  
    struct tnoeud* pFd;  
} TNOEUD;
```

On utilisera un fichier de sauvegarde pour enregistrer les données à la sortie du programme. Au début du programme on récupère les données depuis le fichier dans l'arbre binaire. L'utilisateur est chargé de recevoir les appels téléphoniques des maires et de mettre à jour l'indicateur signature en fonction de ce que dit le maire. Il faut donc pouvoir rechercher rapidement un maire par son nom et mettre à jour l'indicateur signature.

Au final le programme devra offrir les options suivantes à partir d'un menu de sélection :

- (1) créer le fichier des maires,
- (2) récupérer et reconstruire l'arbre/liste à partir d'un fichier.
- (3) rechercher et afficher un enregistrement,
- (4) afficher tous les enregistrements par ordre alphabétique
- (5) modifier la valeur du champ signature,
- (6) sauvegarder les informations dans un fichier.
- (7) sortir du programme

Question 1

1.a_ Faire une fonction de prototype `FILE * fOuvrirFichier(char *nomfic)` qui permet d'ouvrir le fichier de nom `<nomfic>` en lecture écriture. Attention si le fichier existe déjà il doit être ouvert, s'il n'existe pas, il doit être créé.

1.b_ Faire une fonction de prototype `int fCreerEnregistrement(FILE * f)` qui permet de créer un enregistrement (un maire) de type `TNOEUD` et qui l'enregistre à la fin du fichier `f`. On notera que la structure `TNOEUD` utilisée contient deux champs `pFg` et `pFd` qui ne servent à rien pour l'instant et peuvent être mis à `NULL`. On positionnera par défaut l'indicateur signature à `"???"`. La fonction renvoie 1 si l'opération s'est bien déroulée.

- 1.c_ Faire une fonction de prototype `int fCreerFichier(FILE *f)` qui permet de créer une suite d'enregistrements.
- 1.d_ Faire l'option 1 du menu.

Question 2

- 2.a_ Faire une fonction de prototype `TNCEUD * fCreerMaire(FILE *f)` qui à partir du fichier préalablement positionné au bon endroit lit une structure `TNCEUD` et met à `NULL` les champs `pFg` et `pFd`. La fonction renvoie l'adresse du `TNCEUD` récupéré.
- 2.b_ Faire une fonction de prototype `TNCEUD * fInsérerNoeud(TNCEUD * r, TNCEUD * n)` qui à partir d'un arbre existant de racine `r` insère à la bonne place le nœud `n` préalablement créé.
- 2.c_ Implanter la fonctionnalité de récupération des données (depuis un fichier) et de reconstruction de l'arbre binaire (option 2 du menu).

Question 3

- 3.a_ Faire une fonction de prototype `void fAfficherNoeud(TNCEUD *t)` qui affiche les informations d'un nœud quelconque pointé par `t`.
- 3.b_ Faire une fonction de prototype `int fRechercherEnr(TNCEUD *r, char * nomMaire)` qui à partir de l'arbre binaire de racine `r` va chercher un maire de nom `nomMaire` et afficher les informations le concernant. La fonction renvoie `-1` si le nom n'est pas trouvé, `1` sinon.
- 3.c_ Implanter la fonctionnalité dans le programme. Option 3 du menu.

Question 4

- 4.a_ Faire une fonction de prototype `void fAfficherClasse(TNCEUD *r)` qui affiche à l'écran la liste triée des maires avec toutes les informations le concernant par ordre alphabétique.
- 4.b_ Implanter la fonctionnalité (option 4 du menu).

Question 5

- 5.a_ Faire une fonction de prototype `int fModifierEnr(TNCEUD * r, char *nomMaire)` qui à partir de l'arbre de racine `r` va chercher le nœud contenant les informations du Maire de nom `nomMaire` et modifie ses informations. On se contentera de modifier la valeur du champ signature. La fonction renvoie `1` si le noeud a été trouvé, `-1` si il n'a pas été trouvé.
- 5.b_ Implanter la fonctionnalité (option 5 du menu)

Question 6

- 6.a_ Faire une fonction de prototype `int fSauvegardeNoeud(TNCEUD *t, FILE *f)` qui écrit dans le fichier `f` le nœud pointé par `t`. La fonction renvoie `1` si tout s'est bien passé.
- 6.b_ Faire une fonction de prototype `int fSauvegardeArbre(TNCEUD *r, FILE *f)` qui écrit dans le fichier `f` les un à la suite des autres les structures de l'arbre de racine `r`.
- 6.c_ Implanter la fonctionnalité dans le programme en demandant à l'utilisateur le nom du fichier sur lequel il veut enregistrer. (option 6 du menu).

Question 7

Faire une fonction qui désaloue l'arbre créé avant de sortir du programme. Attention n'oubliez pas de demander à l'utilisateur s'il a sauvegardé son travail avant.