Série de TP N°02

Exercice 1:

Ecrire un programme en langage C permettant de créer un tableau de N entiers. Le nombre d'éléments N est donné par l'utilisateur. Grâce au type pointeur et l'allocation dynamique, vous pouvez créer un tableau ayant exactement N éléments (la fonction `malloc` de la bibliothèque <`stdlib.h`>). Le programme doit remplir le tableau avec des valeurs saisies au clavier.

Le programme doit trier les éléments du tableau dans l'ordre croissant (utilisez le tri par permutation). Commencez par la définition d'une fonction *permuter* qui permute les valeurs de deux variables. Le programme affichera ensuite les éléments du tableau trié.

Remarque : utilisez un pointeur pour parcourir le tableau.

Exercice 2:

Nous souhaitons écrire un programme en langage C pour la gestion des employés d'une entreprise. Nous retenons les informations suivantes pour un employé : numéro de l'employé (séquentiel), nom et prénom, la date de recrutement, et le salaire. On suppose que le programme affichera le menu suivant :

```
#####################################################
   1.  Ajouter des employes.
   2.  Mettre a jour les informations d'un employe.
   3.  Supprimer des employes
   4.  Afficher l'employe le plus ancien
   5.  Afficher les employes ayant leurs salaires inferieur
       a une valeur au choix de l'utilisateur.
   6.  Afficher tous les employes de l'entreprise.
   7.  Quitter.
#####################################################
       Votre choix :
```

Travail à faire :

1. Définir les structures de données nécessaires pour implémenter la gestion des employés avec une liste chainée.
2. Pour chacun des choix du menu ci-dessous implémentez une fonction Langage C permettant d'effectuer le traitement correspondant à savoir :
    a. Une fonction *Ajouter* permettant d'ajouter des enregistrements employés à la liste chainée, l'utilisateur doit pouvoir ajouter des employés tant qu'il le souhaite.
    b. Une fonction *mettreAjour* permettant de mettre à jour les informations d'un employé, le numéro de l'employé est lue par la fonction.
    c. Une fonction *supprimer* permettant de supprimer un employé de la liste chainée. Le numéro de l'employé à supprimer est lu dans la fonction. Les numéros des employés qui le succèdent dans la liste doivent être mis à jour.
    d. Une fonction *plusAncien* qui affiche le nom et le prénom de l'employé le plus ancien de l'entreprise. Il est important d'implémenter une fonction pour la comparaison des dates.
    e. Une fonction *afficher_salaire* qui affiche le nom et le prénom de tous les employer ayant leurs salaires inférieurs à une valeur au choix de l'utilisateur qui sera lue par la fonction.
    f. Une fonction *Afficher* permettant d'afficher toutes les informations des employés, une ligne par employé.

Remarque :

- Selon le choix de l'utilisateur, le programme appellera la fonction correspondante.
- Le programme langage C permettant d'afficher le menu est disponible sur le site e-learning, le code inclue la définition des structures de données mais à compléter. Le programme inclue aussi une fonction *`currentDate()`* qui renvoie la date courante du system.

Série de TP N°02

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <time.h>
typedef struct{
int jour,mois,annee;
}Date;
typedef struct{
char nom[50],prenom[50];
}Employe;
Date currentDate(){
Date d;
struct tm *t;
time_t tamp = time(NULL);
t = localtime(&tamp);
d.annee = 1900+t->tm_year;
d.mois = t->tm_mon;
d.jour = t->tm_mday;
return d;}
int main()
{char c='1';
while(c!='o'){
printf("\n\n\n\n");
puts("  ########################################################");
puts("     1.  Ajouter des employes.");
puts("     2.  Mettre a jour les informations d'un employe. ");
puts("     3.  Supprimer des employes");
puts("     4.  Afficher l'employe le plus ancien");
puts("   5. Afficher les employes ayant leurs salaires inferieur \n \t a une
            valeur au choix de l'utilisateur.");
puts("     6.  Afficher tous les employes de l'entreprise.");
puts("     7.  Quitter.");
puts("  ########################################################");
printf("\t Votre choix :   ");  scanf("%c",&c);
switch(c){
case '1': system("cls");puts("\n\t to be completed !!!"); getche(); break;
case '2': system("cls");puts("\n\t to be completed !!!"); getche(); break;
case '3': system("cls");puts("\n\t to be completed !!!"); getche(); break;
case '4': system("cls");puts("\n\t to be completed !!!"); getche(); break;
case '5':system("cls");puts("\n\t to be completed !!!"); getche();  break;
case '6':system("cls");puts("\n\t to be completed !!!"); getche();  break;
case '7':printf("voulez vous quitter le programme o/n ??\n");
                  c =(char)getch(); break;
}
system("cls");
}
    return 0;
}
```