

**Exercice 1:**

On caractérise chaque étudiant par :

- Son *nom* (20 caractères au maximum),
- son *prénom* (20 caractères au maximum),
- son *age* (int),
- son *CNE* (int).

- 1- Définir un type structuré permettant de représenter un étudiant.
- 2- Ecrire une fonction en C++, void creerEtudiant(etudiant \*E1) ou void creerEtudiant(etudiant &E1) qui invite à la saisie un étudiant au clavier.
- 3- Ecrire une fonction en C++, void afficherEtudiant(const etudiant &E1), qui affiche clairement les caractéristiques d'un étudiant reçu en paramètre.
- 4- Ecrire une fonction en C++, bool egales(const etudiant &E1, const etudiant &E2) qui compare les caractéristiques des étudiants E1 et E2. La fonction retourne false si les étudiants ont des caractéristiques différentes, et true si les deux étudiants ont les mêmes caractéristiques.
- 5- Ecrire une procédure void modifieEtudiant(etudiant\*E1) ou void modifieEtudiant(etudiant &E1) qui modifie les caractéristiques d'un étudiant donné.
- 6- Ecrire une procédure .....afficherEtudiants( ....) qui affiche les n étudiants contenues dans un tableau des étudiants,
- 7- Ecrire une procédure .....initEtudiants(....) qui initialise (créer) un ensemble des étudiants comportant n étudiants,
- 8- Ecrire une fonction .....ageMoyEtudiants(....) qui renvoie l' age moyen des d'un ensemble de n étudiants.

**Exercice 2:**

- 1- Donner la déclaration de la structure suivante :  
Date : numero du jour, nom du mois, annee.
- 2- Ecrire une fonction *AfficheDate* affichant la date passée en parametre  
Prototype : void *AfficheDate* (const date &);
- 3- Ecrire une fonction *DemandeDate* demandant une date au clavier et remplissant les champs de la structure passée.  
Prototype : void *DemandeDate* ( date &);
- 4- Ecrire une fonction *NumeroMois* retournant l'entier compris entre 1 et 12 correspondant au numéro du mois dont le nom est passé. Si le mois n'existe pas, cette fonction doit retourner -1.  
Prototype : int *NumeroMois*(char\* momois);  
A l'aide de cette fonction, améliorer la fonction demande date afin qu'elle vérifie que la date entrée est correcte.
- 5- Ecrire une fonction *CompareDate* prenant deux dates d1 et d2 et retournant -1 si d1 avant d2, 1 si d1 est après d2, ou 0 si les deux dates sont égales.  
Prototype : int *CompareDate* (const date &,const date &);