ATELIER 4: ALLOCATION DYNAMIQUE

Exercice 1:

Ecrire un programme qui lit deux tableaux d'entiers **tabA** et **tabB** et leurs dimensions **nA** et **nB** au clavier et qui ajoute les éléments de **tabA** à la fin de **tabB**. Afficher par la suite le contenu de **tabB.**

Exercice 2:

On définie une Personne par un nom (chaine de 20 caractères), un prénom (chaine de 20 caractères) et un âge (un réel). Ecrire un programme qui saisie **n** Personne puis fait l'affichage.

Exercice 3:

On définie une **Personne** par un nom, un prénom et une date de naissance. Le champ date de naissance est un pointeur sur une structure **Date** (jours, mois, année).

Ecrire un programme qui déclare un pointeur sur une variable de type Personne. Faite la saisie puis l'affichage d'une Personne.

La saisie de la date est optionnelle, dans le cas où aucune date n'est fournie on affecte **NULL** au champ de la Date.

Exercice 4:

On définie une **Personne** par un nom, un prénom et une adresse. Le champ adresse est un pointeur sur une structure **Adresse** caractérisée par un numéro, l'avenue et le nom de la ville.

Ecrire un programme qui déclare un pointeur sur une variable de type Personne. Faite la saisie puis l'affichage d'une Personne. La saisie de l'adresse est optionnelle, c'est possible d'avoir plusieurs adresse pour la même personne et dans le cas où aucune adresse n'est fournie on affecte **NULL** au champ de l'adresse.

Exercice 5:

On définie une Personne par un nom, un prénom, un âge et une Personne « enfant ». Le champ enfant est un pointeur sur une structure Personne qui dans ce cas va avoir donc un nom, un prénom et un âge.

Ecrire un programme qui déclare un pointeur sur une variable de type Personne. Faite la saisie puis l'affichage d'une Personne.

La saisie de l'enfant est optionnelle, dans le cas où la personne saisie n'a pas d'enfant on affecte NULL au champ enfant.

Exercice 6 :

1. Ecrire la fonction ***int *Creer(int *T, int n)*** permettant de créer et de saisir n éléments d'un tableau d'entier **T** et de retourner son adresse.
2. Ecrire la fonction ***void Afficher(int *T, int n)*** permettant d'afficher le contenu d'un tableau d'entier **T** composé par **n** cases.
3. Ecrire la fonction ***int * AjoutFin(int *T, int *n, int A)*** permettant d'ajouter un entier **A** à la fin d'un tableau d'entier **T** composé par **n** cases.
4. Ecrire la fonction ***int * AjoutDebut(int *T, int *n, int A)*** permettant d'ajouter un entier **A** au début d'un tableau d'entier **T** composé par **n** cases.
5. Ecrire la fonction ***int * Inserer(int *T, int *n, int p, int A)*** permettant d'insérer un entier **A** à la position **p** dans tableau d'entier **T** composé par **n** cases.
6. Ecrire la fonction ***int *Supprimer(int *T, int *n, int A)*** permettant de supprimer un entier **A** s'il existe à partir d'un tableau d'entier **T** composé par **n** cases.
7. Ecrire un programme permettant de déclarer et de saisir un entier n (n>0) et déclarer un pointeur p sur les entiers et de tester par la suite l'ensemble des fonctions déclarée ci-dessus.

Exercice 7 :

On souhaite réaliser un programme pour la gestion de livres dans une librairie. Un Livre est identifié par un code (entier), un titre (chaîne de caractères), un auteur (chaine de caractères), le nombre d'exemplaires en stock (entier) et le prix (réel). Les livres de la librairie sont stockés dans un fichier.

1. Définir la structure **Livre**.
   Écrire les fonctions suivantes :
2. ***float prixTotal (char chemin[]) ,*** qui permet de calculer la somme des prix de tous les livres du stock.
3. ***int nombreExemplaires (char chemin[], int code) ,*** qui retourne le nombre d'exemplaires pour un Livre dont le code est passé en paramètre. Si le livre n'existe pas, la fonction retourne -1.
4. ***void approvisionner (char chemin[], Livre L)*** qui permet d'ajouter un exemplaire du livre *L* au stock. Si le code correspondant existe déjà, alors le nombre d'exemplaires est incrémenté de 1, sinon on ajoute le nouveau livre à la fin du fichier.
5. ***int nombreTitres (char chemin[]) ,*** qui retourne le nombre de titres dans le fichier.
6. ***int chargerTableau (char chemin[], Livre* Tab),*** la fonction va copier tous les Livres stocker dans le fichier dans le tableau. Ainsi, s'il y a *n* Livres dans le fichier, la fonction va allouer avec *Tab* un tableau de *n* cases, puis elle va parcourir tout le fichier et copier Livre par Livre dans le tableau. la fonction retourne *n*, le nombre de cases remplis dans *Tab*.