Programmation C
TP4 : Allocation dynamique

Pré-requis : chapitres 1 à 9 du cours de Programmation Bas Niveau.

Après un scanf d'un caractère ou d'une chaîne de caractères, il convient d'utiliser la fonction getchar() afin d'éliminer du *buffer* le caractère '\n' entré au clavier lors de l'appui sur Entrée. Dans le cas contraire, le scanf suivant pourrait ne pas capter de caractères.

Exercice 1 : Tableaux de Point2D

1. Écrire une fonction **Point2D * saisieTabPoints2D (int * taille)** qui demande à un utilisateur de saisir, tant qu'il le désire, des coordonnées de points 2D et les ajoute dans un tableau. La fonction renverra ce tableau et modifiera le paramètre taille, passé par référence, afin de communiquer à la fonction appelante la taille du tableau ;
2. Écrire une fonction **void affichePoints2D (Point2D * tab, int taille)** qui affiche tous les Point2D du tableau **tab** passé en paramètre de taille **taille** également passée en paramètre ;
3. Écrire un main et tester la fonction **saisirTabPoints2D** ;
4. Écrire une fonction **Point2D * symetrieOrigine (Point2D * tab, int taille)** qui étant donné un tableau de Point2D **tab** passé en paramètre et sa taille **taille** également passée en paramètre, renvoie un tableau contenant le symétrique de chacun des points de **tab** par rapport à l'origine.
5. Écrire une fonction **int supprimePointsDistanceOrigine (Point2D * tab, int taille, double distance)** qui supprime du tableau de Point2D **tab** passé en paramètre, de taille **taille** passée en paramètre, tous les Point2D dont la distance à l'origine est supérieure à **distance**. La fonction renvoie la nouvelle taille du tableau après suppression des éléments. Le tableau **tab** doit avoir été alloué dynamiquement afin de permettre à la fonction de réajuster sa taille au plus près des besoins avec un realloc. (Attention cet exercice est complexe)

Exercice 2 : Structure Etudiant

1. Définir une structure **etudiant** et son raccourci **Etudiant** qui comprend :
   (a) un pointeur **prenom** vers une chaîne de caractères encodant le prénom ;
   (b) un pointeur **nom** vers une chaîne de caractères encodant le nom ;
   (c) un entier **annee** pour encoder la promotion de l'étudiant.
2. Écrire une fonction initialiseEtudiant qui :
   — demande à l'utilisateur un nom et le stocke dans une chaîne de caractères temporaire d'une taille prédéfinie et assez importante ;
   — demande à l'utilisateur un nom et le stocke dans une chaîne de caractères temporaire d'une taille prédéfinie et assez importante ;

- demande à l'utilisateur un entier pour initialiser la promotion ;
- copie les chaînes temporaires dans les champs **prenom** et **nom** en leur allouant l'espace nécessaire ;
- renvoie l'étudiant créé.

3. Écrire une fonction **void libereMemoireEtudiant (Etudiant e)** qui libère les zones mémoires allouées dynamiquement au sein du paramètre **e**.