TP3 : Structures en C

CSH : Initiation au C et au shell

★ Exercice 1: Gestion de points sur le plan.

L'objectif de cet exercice est de définir un module de programme gérant les points du plan en s'inspirant du design de code donné dans le chapitre 4 du cours au sujet de la programmation modulaire en C.

▷ **Question 1:** Définir une structure de données `point_t` décrivant un point et écrire une fonction constructeur ainsi qu'une fonction destructeur. Vous ajouterez un champ `id` à votre structure, contenant un entier unique identifiant chaque point, entier choisi par le constructeur à la création de l'objet.

▷ **Question 2:** Écrire une fonction qui permet d'afficher les différents champs d'une structure `point_t` avec le format *id(x,y)*. La fonction prend en argument un pointeur vers une structure `point_t`.

▷ **Question 3:** Écrire un ensemble de tests dans un nouveau fichier (qui ne chargera que le .h de votre module) et effectuant quelques actions sur les points que vous définierez. Vous augmenterez cette classe de tests au fur et à mesure de votre avancée dans les questions ci-dessous.

▷ **Question 4:** Définir un *copy constructor*, c'est-à-dire une fonction prenant un pointeur vers une structure `point_t` et créant un nouvel objet étant la copie conforme du premier.

▷ **Question 5:** Définissez et testez la fonction `point_move(point_t *p, double dx, double dy)`.

▷ **Question 6:** Écrire une fonction `symetrique` qui prend en argument un pointeur vers un `point_t` ainsi qu'une option parmi XX,YY,ORIG selon laquelle, la fonction *symetrique* modifie les coordonnées d'un point avec celles de son symétrique par rapport à l'axe des x, des y ou par rapport à l'origine respectivement.

On utilise une énumération pour déclarer les différentes options possibles. La fonction retourne 0 si l'opération se déroule correctement, et un code d'erreur sinon (1 par défaut, mais vous pouvez définir différents code de retour pour différents types d'erreurs).

▷ **Question 7:** Définissez une structure `polygone_t` définie par un ensemble de point constituant ses sommets. On pourra avoir au maximum MAX sommets par polygone (avec MAX défini par un #define dans le code).

▷ **Question 8:** Écrivez un constructeur, un destructeur et une fonction d'affichage pour `polygone_t`. Définissez et testez la fonction `polygone_translate(polygone_t p, double dx, double dy)`.

▷ **Question 9:** Calculez taille de la structure `point_t` avec l'opérateur `sizeof()`. Comparez cette taille à la somme des tailles des différents éléments qui la composent. Que se passe-t-il si le champ `id` est placé devant et devient de type `short` ou `char` ?

▷ **Question 10:** Afin de comprendre comment la structure est stockée en mémoire, écrire une macro OFFSET(x,Y) qui donne pour un élément x de la structure Y, son adresse relative par rapport à l'adresse de début de la structure. Afficher l'offset de chaque élément dans les 2 structures. Conclure !

★ Exercice 2:

On souhaite créer un programme en C gérant un annuaire très simplifié qui associe à un nom de personne un numéro de téléphone.

▷ **Question 1:** Créer une structure `personne_t` pouvant contenir ces informations (nom et téléphone). Le nom peut contenir 32 caractères et le numéro 16 caractères. Écrivez le constructeur, le destructeur et la fonction d'affichage de cette structure.

▷ **Question 2:** Créer une nouvelle structure qui va représenter le carnet d'adresses. Cette structure `carnet_t` contiendra un tableau de 20 Personne et un compteur indiquant le nombre de personnes dans le tableau. Écrivez le constructeur et le destructeur de cette structure.

▷ **Question 3:** Créer une fonction qui ajoute une personne dans un carnet.

▷ **Question 4:** Créer une fonction qui affiche un carnet.

▷ **Question 5:** À partir des étapes précédentes, faire programme gérant un carnet d'adresse. Créer un menu qui propose d'ajouter une nouvelle personne, d'afficher le carnet ou de quitter.

▷ **Question 6:** Quelques extensions possibles :

- Ajout d'une fonction de sauvegarde de l'annuaire dans un fichier : Les données sauvegardées doivent être lues automatiquement au démarrage.
- Taille dynamique de carnet : trouver une solution permettant d'avoir un nombre variable de personnes dans son carnet d'amis, potentiellement supérieur à 20.