

## TP5 - Makefiles et GDB

**Exercice 1:** Écrire dans un fichier `vecteurs.c` contenant :

1. Un type de données `vecteur` qui est une structure possédant un pointeur de flottants (qui contiendrait donc les coordonnées du vecteur), et un char donnant la taille du vecteur.
2. Des fonctions pour le calcul vectoriel : somme des vecteurs ayant la même dimension, multiplication d'un vecteur avec un scalaire, renvoi de la taille d'un vecteur, renvoi du nombre de valeurs non-nulles dans un vecteur, lecture d'un vecteur au clavier (avec allocation de mémoire !), *clone* d'un vecteur (avec allocation de mémoire !).

Écrire aussi le fichier header `vecteurs.h` correspondant.

Puis écrire un programme qui lit plusieurs vecteurs et entiers et fait divers calculs avec (libre à vous d'imaginer le contenu du programme).

Écrire le fichier `makefile` pour la compilation en étapes de vos fichiers, selon le modèle vu en cours.

---

**Exercice 2:** Pour le même programme de l'exo 2, compilez le code avec l'option de débogage, puis suivez pas-à-pas, à l'aide de `gdb`, le comportement de la fonction de calcul de la somme de deux vecteurs sur des vecteurs de dimension 3 saisis au clavier.

Pour cela, placez un breakpoint avant l'appel de votre fonction "*somme*", puis exécutez pas-à-pas les instructions de celle-ci et affichez, après chaque nouvelle instruction exécutée, d'abord la valeur du pointeur représentant le début du tableau, puis le contenu du tableau représentant la somme des deux vecteurs.

---

**Exercice 3:** Prenez le code du programme qui emploie la fonction de suppression d'un noeud dans une liste chaînée. Compilez ce programme avec l'option de débogage, puis regardez le fonctionnement de votre programme en vous aidant de `gdb`.

Plus précisément, en supposant que votre fonction utilise la variable `p` pour parcourir la liste chaînée et une deuxième variable `prev` pour retrouver le noeud juste avant le noeud courant, affichez l'adresse des noeuds, les entiers stockés dans chaque noeud et le contenu du lien `ptr`. Enfin, lorsque votre fonction arrive aux opérations de suppression de noeud, affichez le résultat de chaque instruction (y compris de `free`) sur l'état des variables.

Modifiez votre fonction de sorte qu'elle *ne vérifie pas* si la position à supprimer est dans la liste, et, à l'aide de `gdb`, notez le fonctionnement de la fonction lorsque vous l'appellez en lui demandant de supprimer le 5<sup>e</sup> élément d'une liste ne contenant que 3.

---