

TP sur le chapitre 4 (types complexes)
--

Exercice 1 : Écrivez un programme qui définit une union de trois réels, une struct de trois réels, et affiche la taille de chacun de ces types (avec l'opérateur `sizeof`). Expliquez la différence de taille.

Exercice 2 : Une personne a un prénom (d'au plus 10 caractères), un nom (d'au plus 10 caractères), et une adresse (d'au plus 20 caractères). Écrivez le type `personne`.

Exercice 3 : Déclarez un tableau de 10 personnes, et initialisez-le avec des personnages fictifs.

Exercice 4 : Écrivez une fonction qui affiche toutes les personnes du tableau.

Exercice 5 : Écrivez une fonction qui détermine s'il existe plusieurs personnes du tableau ayant la même adresse.

Exercice 6 : Écrivez une fonction qui affiche les personnes de la dixième à la première.

Exercice 7 : Écrivez une fonction qui inverse les éléments du tableau.

Exercice 8 : Écrivez une fonction qui affiche les prénoms dans l'ordre lexicographique, sans modifier le contenu du tableau.

Exercice 9 : Écrivez un programme qui déclare un entier, affiche son adresse et sa valeur, modifie sa valeur, et affiche à nouveau son adresse et sa valeur.

Exercice 10 : Écrivez une fonction qui prend en paramètre deux entiers, et échange leurs valeurs. Affichez la valeur des deux entiers avant l'appel, pendant l'appel et après l'appel de la fonction.

Exercice 11 : Écrivez une fonction qui prend en paramètre deux pointeurs sur des entiers, et échange la valeur des entiers pointés. Affichez la valeur des deux entiers avant l'appel, pendant l'appel et après l'appel de la fonction.

Exercice 12 : Écrivez un programme qui demande à l'utilisatrice de saisir un entier n , alloue un tableau de n entiers, stocke dans ce tableau les entiers de 1 à n , affiche ce tableau, et le libère.

Exercice 13 : Écrivez un programme qui demande à l'utilisatrice deux entiers n et m , crée deux matrices de taille $n \times m$, les initialise, les multiplie dans une troisième matrice, et libère les trois matrices ainsi créées. Chaque tâche sera écrite dans une fonction (`saisieDeuxEntiers`, `creationMat`, `multiplicationMat`, `liberationMat`).

Exercice 14 : Si l'outil `valgrind` est installé, utilisez-le sur le programme précédent pour détecter des fuites de mémoire éventuelles.