

Info 2 PC

TP 1

L'objectif de ce TP est d'apprendre à utiliser en pratique ce qui a été vu en Cours et TD. L'architecture Arduino étant plus minimaliste que celle d'un ordinateur, certains aspects seront plus simples par la suite.

Pour tester vos programmes, créez un main et testez vos fonctions/procédures aussitôt écrites, n'attendez pas d'avoir tout écrit avant de tester.

Rappel : la commande de compilation que vous utiliserez est `g++ fichier.cpp -o executable`.

Exercice 1 : Manipulation de structures

1. Dans un fichier `main.cpp`, créez une structure `etudiant`, qui sera composée d'une chaîne `nom` d'au plus 20 caractères, d'un entier `numero_etudiant`, et d'un réel, `noteInfo2`.
2. Créez une fonction permettant de saisir les informations relatives à un `etudiant`, et qui retourne cet `etudiant`.
3. Créez une procédure prenant en entrée un `etudiant` et permettant l'affichage des informations concernant cet `etudiant`.

Exercice 2 : Manipulation de pointeurs

1. Dans votre `main`, déclarez un entier `n` et un pointeur `p` sur `etudiant`.
2. Demandez à l'utilisateur de saisir le nombre `n` de notes d'`etudiant` à saisir.
3. Allouez de la mémoire à votre pointeur `p`.
4. Créez une fonction `saisirNotes` prenant en entrée ce pointeur `p` et l'entier `n`, et qui demande à l'utilisateur de saisir les notes des `etudiants`.

Exercice 3 : Manipulation de fichiers

1. Créez une fonction `enregistrerEtudiants` qui prend en entrée un pointeur sur `etudiant` et un nom de fichier. La fonction ouvre le fichier en écriture (pensez à le fermer), écrit les informations avec le formatage suivant :
`nom_1 numero_etudiant_1 noteInfo2_1`
`nom_2 numero_etudiant_2 noteInfo2_2`
...
2. Créez une fonction `lireEtudiants` qui prend en entrée le nom du fichier, l'ouvre en lecture seule (pensez à le fermer) et retourne un pointeur sur `etudiants` contenant l'ensemble des données du fichier. (Ne vous lancez pas tête baissée dans cet algo, réfléchissez ! De quoi avez-vous besoin pour allouer juste ce qu'il faut de mémoire à votre pointeur ? Comment le faire ? ...)

Exercice 4 : Création d'un algo de tri

1. Rendez-vous à la page 31 du cours sur la mémoire dynamique [<http://www-perso.unilim.fr/deneuille/docs/Info2PC/Cours3.pdf>] et étudiez la procédure de tri par nombre d'habitants.
2. Adaptez-la pour trier des `etudiants` par ordre croissant de leur note `noteInfo2` et traduisez la en `C++`.
3. Écrire une fonction `trierEtudiant` qui prend en entrée le nom du fichier dans lequel sont stockées les informations relatives à des `etudiants`, tri ces `etudiants` par ordre croissant de leur note, et sauvegarde le résultat dans un fichier texte `etudiant-tries.txt`.

L'environnement de développement que nous utiliserons pour programmer sur Arduino intègre déjà un compilateur. Si vous n'aurez plus besoin de retenir la syntaxe exacte pour rendre votre petit bout de programme exécutable, les erreurs quand à elles resteront les mêmes. C'est pourquoi je vous incite vivement à essayer de comprendre ce que vous dit le compilateur dès maintenant.