

TD 6 - Outils de programmation

Techniques de Programmation

Exercice 1

1. Télécharger le fichier `article.h`. Il définit le type `Article` contenant les informations propres à un article, à savoir son code, sa désignation, sa taille et son prix hors taxe.
2. Dans un fichier `article.c` créer une fonction `int (Article ** article, int iCode, char * sDesignation, int iTaille, float fHT);`. Cette fonction allouera dynamiquement un article à partir de ses paramètres. Créer ensuite une fonction `void detruireArticle (Article ** article);` libérant l'espace mémoire alloué pour un article.
3. Dans un fichier `gestion.c`, créer un programme permettant d'ajouter et de supprimer des articles dans un tableau de pointeur.
4. écrire un `makefile` permettant de compiler ces fichiers et de créer un exécutable.

Exercice 2

Dans cet exercice, la gestion de la mémoire doit être dynamique pour toute structure, chaîne de caractère ou tableau. Le programme sera conçu en respectant les préceptes de la programmation modulaire et de la programmation défensive. Il va donc falloir créer une structure de données `Etudiant` (ainsi que ses fonctions) et un programme principal la manipulant.

Afin d'effectuer une photo de promotion, on souhaite écrire un logiciel permettant de trier les étudiants en fonction des critères retenus par le photographe.

Pour cela, créez un structure de donnée permettant de manipuler un étudiant, caractérisé par son nom, son prénom et sa taille.

Créer ensuite un programme permettant de manipuler un ensemble d'étudiants dont les informations sont stockées dans un tableau dynamique. La taille du tableau sera donnée au lancement du programme en paramétrant le programme (utiliser : `int main(int argc, char * argv[])`). Le premier paramètre du programme sera le nombre d'étudiant. Le second sera l'ordre du tri. Croissant ou décroissant optionnel, croissant par défaut).

Pour effectuer le tri, étant donné que les structures sont complexes, il vaut mieux créer une fonction spécifique dans la structure de données `Etudiant` effectuant une permutation de deux instances d'étudiant.

Le programme effectuera la saisie du tableau, son affichage avant le tri, son tri, et son affichage après le tri. Attention à bien libérer la mémoire !

Exercice 3

L'objectif de cet exercice est d'apprendre le maniement du débogueur `ddd`.

Écrire un programme qui calcule la somme des `n` premiers entiers. compiler le programme avec l'option `-g`. Lancer `ddd` avec comme paramètre le nom de l'exécutable.

L'interface s'ouvre avec :

- le fichier qui contient le main dans le cadre du haut ;
- la console de commandes `gdb` dans le cadre du bas ;
- la fenêtre de commandes `ddd` (command tool).

Pour afficher la console d'exécution, sélectionner l'item `Execution Window` dans le menu `View`. On obtient alors une fenêtre intitulée `Execution window`. Pour afficher le cadre des variables en haut de la fenêtre principale, sélectionner l'item `Data Window` dans le menu `View`.

Pour placer un *point d'arrêt*, Cliquer sur la ligne où vous souhaitez placer le point d'arrêt puis cliquer sur l'icône représentant un stop. Pour lancer l'exécution, dans la fenêtre DDD, cliquer sur **Run**. Une flèche verte apparaît pour montrer l'endroit de l'exécution (sur le breakpoint).

Pour afficher la valeur d'une variable, un double click sur une variable dans la fenêtre contenant le code, affiche les variables et leur valeur dans le cadre du haut, il est possible de déplacer chaque représentation de la variable et de son contenu. NB : si le double click ne fonctionne pas, vous pouvez vous positionner dans le texte sur une variable et faire un click avec le bouton droit, un menu apparaît et vous pouvez choisir **display**. Lorsqu'on clique avec le bouton droit de la souris sur une des variables dans le cadre du haut, un menu s'affiche, il permet par exemple de supprimer une variable du cadre (**undisplay**), de changer le type d'affichage, d'affecter une valeur à la variable (**set value**).

Pour se déplacer d'un breakpoint à un autre, ou pas à pas, il suffit de cliquer dans la fenêtre DDD sur le bouton **Next** pour se déplacer d'instruction en instruction, ou sur le bouton **Cont** pour se déplacer jusqu'au prochain breakpoint.

Télécharger le fichier source `exobogue.c.tar.gz` sur le site et corriger le pour qu'il se termine sans erreur.