TP5 : Tris et récursivité

Séances 8 et 9 de TP de C – Evaluation du binôme en séance 9

1. Objectifs

L'objectif de cette séance est d'appliquer, en langage C, les algorithmes sur les tris qui seront étudiés en cours d'algorithmique. Vous aurez donc à coder une bibliothèque permettant l'utilisation de tris pour des tableaux d'entiers.

Les notions de langages C qui vous seront utiles sont les tableaux, les fonctions ainsi que les pointeurs sur tableaux, et les pointeurs sur fonctions. L'utilisation de la bibliothèque sur les nombres aléatoires sera aussi mise en œuvre.

Vous développerez cette bibliothèque dans des fichiers séparés afin de rendre votre code réutilisable. Chaque fichier sera consacré à un tri et un seul. Un fichier sera consacré au remplissage aléatoire du tableau d'entiers et à son affichage. Vous devrez mettre en place des tests unitaires (avec « seatest ») qui vous permettront de vérifier le bon comportement de vos fonctions.

L'utilisation de la fonction « scanf » est de nouveau interdite, même durant la phase de développement et de mise au point.
Vous devrez donc concevoir vos tests unitaires en même temps que vos fonctions.

2. Les fonctionnalités attendues

Ci-dessous est listé l'ensemble des fonctionnalités que devra fournir votre bibliothèque. Chacune de ces fonctionnalités possédera un ensemble de tests unitaires.

Le temps de travail est indiqué à titre informatif. Il correspond au temps qu'il devrait vous falloir pour coder chaque partie.

1° partie : Tris simples

Tout cela a déjà été codé en TD d'algorithmique (Voir le site dev.isen). Ces tris s'appliquent pour des tableaux simples d'entiers

- Ecrire une fonction qui fait un tri bulle
- Ecrire une fonction fait un tri par insertion
  - avec une recherche dichotomique simple
  - avec une recherche dichotomique récursive.
- Dans les 2 tris précédents, remplacer les comparaisons par une fonction, que vous devrez écrire. Cette fonction masque le traitement sur la relation d'ordre entre 2 éléments a et b du tableau. C'est-à-dire que la fonction devra rendre :
  - Une valeur négative quand a < b
  - Une valeur égale à 0 quand a = b
  - Une valeur positive quand a > b

Proposer un jeu d'essais pertinent pour les tests unitaires

 45 minutes pour les fonctions + 20 minutes pour les tests unitaires

2° partie : Le tri rapide

Ce tri a été étudié en TD d'algorithmique. Il s'agit donc d'une simple traduction de simple langage vers C

- Ecrire une fonction qui implémente le tri rapide vu en cours d'algorithmique.
- Utiliser la fonction de tri précédemment écrite dans ce tri.

Proposer un jeu d'essais pertinent pour les tests unitaires

 1 heure pour la fonction + 30 minutes pour les tests unitaires

3° partie : Fonction « qsort » et relation d'ordre.

Mise en œuvre de la fonction « qsort »

- Etudier et utiliser la fonction qsort pour trier un tableau d'entiers.
- Créer une nouvelle fonction de tri avec le format attendu, qui mettra en œuvre la relation d'ordre suivante :
  - Les nombres impairs sont inférieurs aux nombres pairs.
  - Les nombres impairs sont triés par ordre croissant.
  - Les nombres pairs par ordre décroissant.
- Utiliser cette fonction de tri dans qsort.

Réutiliser le jeu d'essais et les tests unitaires de la partie précédente

 1 heure pour la mise en œuvre de « qsort », 1 heure pour la mise en oeuvre de la fonction de tri et 30 minutes pour les tests unitaires

3. Astuces

- Démarrez un nouveau projet – Ne repartez pas d'un ancien TP
- Aidez-vous de vos TD d'algorithmique
- Réutilisez ce que vous avez fait lors du TP précédent sur les tests unitaires
- Concevez vos tests unitaires avant de coder les fonctions testées

4. Evaluations

- Maîtrise de Netbeans (Création d'un projet, ajout de fichiers sources .c/.h, compilation, debugger)
- Conventions de codage (commentaires, indentation, nom des fichiers, des fonctions, des variables, ...)
- Compilation séparée .h/.c, séparation du code en fonctions
- Tableaux, boucles
- Pointeurs, passage de paramètres par valeur/par adresse
- Respect des contraintes de l'énoncé
- Tests unitaires & jeux d'essais