GUIDE DES BONNES PRATIQUES POUR LES TP DE SDD EN C

I. GÉNÉRAL

Notes : les dossiers (et le code) sont notés, le coefficient de chaque TP est égal à sa durée (ex : le TP1 est coef 4)

Absences : toute absence non justifiée est sanctionnée par soustraction de point sur la note (ex pour le TP1, 1 abs=-5 pts, pour 1 TP de 2 semaines 1abs=note/2)

Utiliser obligatoirement un debugger pour la mise au point.

Apporter en TP le polycop de C (voir en particulier en annexe les formats de `scanf` et `printf` nécessaires pour le TP1 + fonctions `strncpy, strncmp`) et les TD de SDD.

Interdit de saisir des données au clavier (perte de temps). Tous les tests (même les premiers) doivent être réalisés sur des fichiers.

II. CODAGE

Respecter les critères de qualité de l'algorithmique :

- découpage en fonctions (1 fonction = 1 action sur 1 SDD) réparties en modules (regrouper dans un module les fonctions logiquement reliées).
- 1 seul return par fonction
- pas de break dans les boucles (variable booléenne pour arrêter)
- nommage significatif des variables
- pas de variable globale

Compilation séparée et rédaction d'un makefile obligatoire

Faire obligatoirement un fichier .h (1 par SDD en ppe) contenant les définitions de types et les prototypes de fonctions.

Donner pour chaque fonction :

- un lexique complet (pas seulement les paramètres d'entrée/sortie mais aussi toutes les variables locales).
- un exposé rapide du principe en français (en entête) et des commentaires explicites.

Pas d'écritures de messages d'erreur dans les fonctions de base : la fonction retourne un indicateur ou code d'erreur, que le programme appelant interprète et traite.

Jamais de duplication de code, dès qu'un traitement apparaît 2 fois faire une fonction.

Pas de menu long et répétitif. Utiliser des fichiers ou des programmes de tests (qui seront conservés) pour les jeux d'essais. Les noms de fichiers sont donnés en ligne de commande. Organiser les essais de manière à tester rapidement tous les cas mais pas plusieurs fois le même cas !

Pas de données en dur dans le programme (saisir toutes les données, par ex. les valeurs à chercher, etc.)

Ne pas utiliser une variable qui soit à la fois une donnée et un contrôle, en particulier en retour de fonction.

Exemple :

NON : une fonction retourne un entier, si cet entier vaut 0 cela veut dire qu'il a eu une erreur.

OUI : une fonction fournit un entier en paramètre de sortie et retourne un code erreur (1 si erreur).

Cas particulier : on peut concevoir des fonctions qui retournent une adresse ou NULL.

III. DOSSIER

Faire le dossier par 2.

Le modèle de dossier fourni en exemple a été volontairement écrit en texte brut mais on peut utiliser les outils de traitement de texte et de dessin si l'on veut.