TP13 : Erreurs en C

Objectifs du TP :

- Lire et comprendre du code en C.
- Identifier, comprendre et corriger des erreurs dans ce langage.
- Réviser la structure d'arbre binaire de recherche.

Les codes sources à corriger sont dans le répertoire TP13_compagnon, à copier dans votre espace personnel.

1. On considère le code source 0_acces.c.
   a) Le compiler avec la commande `gcc 0_acces.c`. L'exécuter et expliquer.
   b) Le compiler avec la commande `gcc -fsanitize=address 0_acces.c`. Comprendre et corriger.
2. On considère le code source 1_local.c.
   a) Le compiler avec la commande `gcc 1_local.c` et l'exécuter sans précaution.
   b) Comprendre le problème et le corriger. Vous pouvez au passage observer les informations données via `-fsanitize=address` sur ce code source.
3. On considère le code 2_pgcd.c.
   a) Le compiler et l'exécuter. Que constate-t-on ?
   b) Expliquer le comportement obtenu.
   c) Écrire une fonction de calcul de PGCD correcte.
4. On considère le code 3_liberation.c.
   a) Inférer le comportement attendu des fonctions présentées et l'affichage qu'on devrait obtenir.
   b) En s'aidant d'options de compilations adaptées, corriger ce code.
5. On considère le code 4_ub.c.
   a) Le compiler sans option de compilation et l'exécuter.
   b) Est-on satisfait ? Expliquer le problème.
6. On considère le code 5_chaines.c.
   a) Le compiler, l'exécuter et observer le résultat.
   b) Corriger le code tant que la compilation avec `-fsanitize=address` se fait avec erreur.
7. On considère le code 6_listes.c.
   a) Pour chacune des fonctions de ce code source, inférer quelle devrait en être la spécification.
   b) Compiler, exécuter, expliquer le comportement obtenu et corriger le code.

Dans la suite, on considère un morceau de code (dans le répertoire 7_ABR) un peu plus conséquent permettant de manipuler des arbres binaires de recherche dans l'optique de gérer un inventaire de matériel informatique. Il se décompose en :

- Un module base implémentant un type `donnee`. Un objet de ce type est constitué d'une chaîne de caractères correspondant au nom d'un article et d'un entier correspondant à son prix. Dans ce module les fonctions sont correctes. L'une permet d'afficher une donnée, l'autre de les comparer (via une comparaison sur les prix).
- Un module arbre qui implémente des arbres bianires de recherche.
- Un fichier principal main.c destiné aux tests.

Le Makefile fourni permet de compiler les différents fichiers impliqués.

8. Tant que la compilation ou l'exécution se produit avec erreur ou warning, comprendre les erreurs et les corriger. Vous pouvez réécrire complètement une fonction sans chercher à calquer le code proposé du moment que les protoypes et les spécifications des fonctions sont respectées.
9. (bonus) Implémenter la recherche et la suppression dans un ABR pour compléter le module arbre.