

Institution : Institut Grasset
Professeur : Rébecca Lapointe
Cours : Contrôle de la qualité I (420-PA4-AG)
Session : Automne 2017

Descripteur : Travail pratique 3

Objectif : Ajout de fonctionnalité à un système existant.
Encapsulation, allocation de mémoire, et tests unitaires.

Pondération : 35 % de la note finale

Description : Ajouter des fonctionnalités à un noyau programmé en C. En équipe de 2, ou seul.

Mise en situation :

L'engin qui a téléporté Plin L'Ancien à l'époque moderne fait encore des siennes. Kidinnu de Babylone (base 60), Donald Knuth (base 8), Ada Lovelace (base 2) et vous (base 10) vous retrouvez dans un café.

Programmez un petit exécutable en C qui permet aux personnages d'entrer un nombre dans leur base à la console et qui traduit le nombre entré dans les bases désirées par tous les convives.

Tant que l'utilisateur entre une base valide (entre 2 et 60), l'exécution se poursuit. Dès qu'il entre une base négative, nulle, ou plus grande que 60, le programme s'arrête.

Votre programme doit être capable de traduire tout nombre entier dont la valeur en décimal est entre 1 et 9 223 372 036 854 775 807 inclusivement. (Ce sont les valeurs qui peuvent entrer dans un long.)

Voici un exemple d'exécution :

```
$ gcc *.c && ./a.out
Quelle est la base?
10
Entrez un nombre dans la base 10
1025
Nombre entré : 1025
    Ada (base 2) : 10000000001
    Donald (base 8) : 2001
    Vous (base 10) : 1025
Kidinnu (base 60) : H5

Quelle est la base?
2
Entrez un nombre dans la base 2
11111111111111111111111111111111
Nombre entré : 268435455
    Ada (base 2) : 11111111111111111111111111111111
    Donald (base 8) : 1777777777
    Vous (base 10) : 268435455
Kidinnu (base 60) : KgjOF

Quelle est la base?
0
Bye
$
```

Vous est fourni :

Le code qui lit et écrit à la console. Ceci inclut
La fonction `int main()`,

La fonction `void printNombre(Representation * nb)` qui imprime à la console une représentation

La fonction `Representation traduireEntree(char* chaine, int base)` qui transforme une chaîne de caractères écrite en base `base` en une représentation.

La structure `Representation` qui contient tous les attributs d'une représentation.

La fonction `Representation constructeur(int taille, int base)` qui bâtit une représentation vide.

Vous devez programmer :

Le code de la fonction `Representation decimalABase(int base, long valeur)` qui prend une valeur et bâtit sa représentation dans la base désirée.

Le code de la fonction `long baseADecimal(Representation* nb)` qui prend une représentation et retourne sa valeur décimale.

Les tests unitaires pour ces deux fonctions. Ceux-ci sont appelés dans le `main` avant l'exécution du reste du code.

Détail de la remise

Votre projet doit être bien structuré, avec des fichiers d'en-tête et les inclusions nécessaires. Il ne doit contenir qu'un seul `main`.

Chaque fonction doit être commentée selon les conventions vues au premier cours (en incluant vos noms dans chaque fichier). Vous devez également respecter les conventions de nomenclature.

Vous devez remettre tous les `.c` et `.h` nécessaires au bon fonctionnement de votre projet et uniquement ceux-ci.

Votre fichier compressé doit être nommé `NomPrenom_NomPrenom_TP2.tar.gz` ou `NomPrenom_NomPrenom_TP2.zip`. Remettez un seul fichier sur Léa par équipe.

Critères d'évaluation :

- | | |
|---|-------|
| • La traduction de décimal à une base quelconque est correcte | 6 pts |
| • L'algorithme implémenté est efficace et proprement programmé | 6 pts |
| • La traduction d'une base quelconque à décimal correcte | 6 pts |
| • L'algorithme implémenté est efficace et proprement programmé | 6 pts |
| • Les modifications au code fourni et à la structure de données sont bien documentées | 3 pts |
| • Les tests unitaires sont bien conçus | 4 pts |
| • Les tests unitaires sont complets | 4 pts |
| • Le programme est découplé des tests et a une structure robuste | 3 pts |
| • Respect des normes de documentation et de nomenclature | 1 pt |
| • Respect de la méthodologie de la livraison finale | 1 pt |

Total : 40 points (qui seront ramenés sur 35 %)

Remise : Période 15

Politique de retard :

À partir de la PIEA, pour l'enseignement aux adultes, la politique de retard dans les remises de travaux a été précisée de la manière suivante :

- L'étudiant qui remet un travail en retard est pénalisé
- Le professeur enlèvera 10% du maximum des points par jour de retard, incluant les jours de fin de semaine.
- Après la remise des travaux corrigés, les retards ne sont plus admissibles. L'étudiant obtient la note zéro.

Politique de plagiat

Sera considéré comme du plagiat toute utilisation ou reproduction de code en entier ou en partie, provenant d'internet, de manuels de références, de travaux d'autres étudiants ou d'autres cours. **Tout plagiat ou toute fraude, toute tentative de plagiat ou de fraude, toute coopération à un plagiat ou à une fraude entraînent la note zéro pour l'activité concernée.** L'utilisation de code fourni en classe n'est pas considérée comme du plagiat.

Politique linguistique

Le travail doit être produit en langue française (commentaires et documentation).

Politique de contenu

Sont à proscrire tous les contenus à caractère irrespectueux, diffamatoire ou explicite, qu'ils soient sexuels, racistes, religieux ou autres. Tout contenu devra être validé par le professeur avant d'être diffusé.