

Licence informatique 2^{ème} année

Université de La Rochelle

Programmation C - Projet



Christophe Demko <christophe.demko@univ-lr.fr>

2018-2019__1

1 Consignes

1.1 Groupes

Le projet se fait par groupe de 8 à 12 étudiants. La date limite d'inscription est fixée au 22 octobre 2018 (par l'intermédiaire de la fiche d'inscription jointe).

1.2 Langue utilisée

La langue utilisée dans le code et la documentation devra être exclusivement l'anglais. Si vous avez des difficultés dans la langue de Shakespeare, vous pourrez utiliser les traducteurs automatiques :

- <https://www.deepl.com/translator>
- <https://translate.google.fr/>

1.3 Nommage

1.3.1 Fichiers

- les noms de fichiers du langage C seront tous en minuscules et en anglais. S'ils sont composés de plusieurs mots, ils devront être séparés par un tiret (-) ;
- les fichiers contenant le code devront avoir l'extension `.c` ;
- les fichiers d'en-têtes (exportables) devront avoir l'extension `.h` ;
- les fichiers destinés à être inclus dans votre code mais non exportables devront avoir l'extension `.inc`

*© 2017-2018 Christophe Demko. Ce document est distribué sous la licence CC-by-nc-nd (<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.fr>)

1.3.2 Types

Les noms de types devront faire commencer chaque mot qui les compose par une majuscule. Il n'y a pas de sous-tirets. Les structures et les énumérations devront commencer par un sous-tiret (`_`) pour ne pas les confondre avec les noms de types.

1.3.3 Macros

Les macros (avec ou sans arguments) s'écrivent tout en majuscule en séparant les mots par des sous-tirets (`_`).

1.3.4 Variables

Les variables s'écrivent toutes en minuscules en séparant les mots par des sous-tirets.

1.4 Style

1.4.1 Marque d'inclusion unique

Chaque fichier d'en-tête devra posséder une marque permettant d'éviter les conséquences d'un fichier inclus plusieurs fois. Voir https://google.github.io/styleguide/cppguide.html#The__define_Guard

1.4.2 Ordre des inclusions

L'inclusion des fichiers d'en-tête devra respecter la logique suivante :

1. Inclusion du fichier directement lié au fichier `.c` qui l'inclut suivi d'une ligne vide ;
2. inclusion des fichiers d'en-tête du C standard suivis d'une ligne vide ;
3. inclusion des fichiers d'en-tête provenant d'autres bibliothèques suivis d'une ligne vide ;
4. inclusion des fichiers d'en-tête du projet suivi d'une ligne vide ;
5. inclusion des fichiers d'inclusion (extension `.inc`)

1.4.3 Indentation

Le style d'indentation devra être celui préconisé par Google <https://google.github.io/styleguide/cppguide.html#Formatting>. L'utilitaire `clang-format` (<https://clang.llvm.org/docs/ClangFormat.html>) supporte le style Google.

Vous pourrez utiliser l'utilitaire `cclint` pour vérifier votre code.

1.5 Champs protégés

Les champs des structures seront protégés à la manière de la librairie `fraction` vue en travaux pratiques.

1.6 Documentation

La documentation sera générée avec l'outil `sphinx` et les fonctions seront documentées avec la norme de `doxygen`.

1.7 Tests

Des tests unitaires devront être implémentés, ils testeront chaque fonction et s'efforceront de vérifier que la mémoire est bien libérée au moyen de l'utilitaire `valgrind`.

Vous pourrez vous inspirer du projet <https://github.com/chdemko/c-test>. Pour l'installer sur les machines virtuelles étudiantes, vous aurez besoin avant toute chose d'exécuter :

```
$ export HOME="/media/Qi/$USER"  
$ export PATH="/media/Qi/$USER/.local/bin:$PATH"
```

avant les instructions décrites dans le projet.

D'une manière générale, toutes les options possibles décrites dans ce projet devront être implémentées.

2 Sujet

Le but du projet est de produire un logiciel capable de :

- donner les mots rimant avec un mot (en les classant par ordre de richesse de rimes) ;
- produire un système de phonétisation d'un nouveau mot.

Le fichier `Lexique382.csv` (issu du projet <http://www.lexique.org/telLexique.php>) donne pour chaque mot fourni :

- sa syllabisation
- sa phonétisation
- la syllabisation de sa phonétisation

Les quatre champs sont séparés par une tabulation. Attention, il est possible que le fichier comporte quelques erreurs (que vous devrez détecter).

Le projet devra fournir une documentation produite avec

```
$ make docs
```

Il devra comporter une interface graphique implémentée à l'aide de `gtk` et il pourra être installé avec

```
$ make install
```

Historique des modifications

2018-2019_1 *Mardi 2 octobre 2018*

Dr Christophe Demko <christophe.demko@univ-lr.fr>

— Version initiale