Examen - ING1 GM
Programmation C

Modalités

- Durée : 3 heures
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document papier n'est autorisé.
- Aucune sortie n'est autorisée avant une durée incompressible de 1 heure.
- Aucun déplacement n'est autorisé.
- **Aucune question au professeur n'est autorisée**. Si le sujet ne vous semble pas clair, à vous d'expliquer dans des commentaires pourquoi est ce que ça ne vous semble pas clair, et quelles modifications vous effectuez pour réaliser l'exercice.
- Aucun échange, de quelle que nature que ce soit, n'est possible.
- Les différentes fonctions/procédures demandées seront toutes à tester dans un **main**.
- Vous traiterez chaque exercice dans un dossier différent.
- Le barème est indicatif, il intègre pour chaque question un test dans un main.
- La présence d'un Makefile et des commentaires doxygens sont un plus, et seront appréciés.
- La lisibilité du code et les commentaires seront pris en compte dans la notation, de même que la séparation des différentes procédures/fonctions dans des fichiers d'en-tête (.c .h).
- **Tout code qui ne compile pas ne sera pas corrigé et entraînera une pénalité de 50%. Si une partie de votre code ne compile pas, commentez-le. Ceci implique évidemment que vous compiliez au fur et à mesure!**
- Chaque **warning** de compilation entrainera une **pénalité de 15%**.
- Tout ce qui sera dans le dossier rendu correspondra à votre rendu, et sera récupéré à la fin de l'examen. Je vous conseille vivement de travailler directement dans ce dossier.

Exercice : Le pendu (1.5 + 3 + 3 = 7.5)

Le *Pendu* est un jeu consistant à trouver un mot en devinant quelles sont les lettres qui le composent. Le joueur a le droit à 10 propositions mauvaises avant de se faire pendre (et de perdre).
Exemple d'exécution (partielle):

```
Mot : _________         Nb d'essai restant : 10
Quelle lettre voulez-vous jouer ?   A
Mot : _________         Nb d'essai restant : 9
Quelle lettre voulez-vous jouer ?   E
Mot : ____E___E         Nb d'essai restant : 9
Quelle lettre voulez-vous jouer ?   I
Mot : ____E___E         Nb d'essai restant : 8
Quelle lettre voulez-vous jouer ?   O
Mot : __O_E___E         Nb d'essai restant : 8
```

1. Écrire la fonction `int joueLettre (char* motATrouver, char* motMystere, char lettre)` qui prend en paramètre `motATrouver` le mot qui doit être deviné par le joueur, `motMystere` le mot qui est affiché à l'utilisateur et `lettre` la lettre que l'utilisateur veut jouer, qui vient faire apparaître dans `motMystere` la `lettre` donnée et qui renvoie le nombre de fois que la `lettre` est apparu dans le `motMystere`.
2. Écrire la procedure void jouePendu (char* motATrouver) qui prend en paramètre le mot à faire deviner et qui déroule le jeu. Les différentes étapes du jeu sont :
    a. Faire une copie du mot et remplacer tous les caractères de la copie par un caractère marquant la non-découverte de la lettre
    b. Tant que tout le mot n'a pas été découvert et qu'il reste encore des essais
        i. Demander une lettre
        ii. Jouer cette lettre
    c. Annoncer si le joueur a gagné (en combien de coups) ou s'il a perdu.
3. Afin de faire varier les mots à deviner, le mot à trouver sera tiré aléatoirement dans un fichier. Le fichier aura en première ligne le nombre total de mots qu'il contient, puis un mot par ligne.
   Écrire la fonction `char* motAleatoire (char* nomFichier)` qui renvoit un mot prit aléatoirement dans le fichier.

Remarques :

- ➢ Un fichier "mots.txt" est fourni dans le dossier `docs/` contenant une liste de mots prédéfinis.
- ➢ Générer un nombre aléatoire se fait à travers la fonction `rand()` qui renvoie un entier entre 0 et `RAND_MAX`. Pour limiter le nombre aléatoire à une certaine valeur, il suffit de faire le modulo. Afin d'obtenir des nombres aléatoires différents, il faut "initialiser" la fonction `rand`. Pour cela, il faut inclure la librairie `time.h` et, au tout début du main, faire l'instruction `srand(time(NULL));`

Exercice : Fractions (0.5 + 1 + 1 + 2 + 4 = 8.5)

1. Définir la structure `Fraction` qui est composée d'un numérateur et d'un dénominateur entiers.

2. Écrire la fonction `saisirFraction` qui demande la saisie du numérateur et dénominateur à l'utilisateur et qui renvoie une variable de type `Fraction`.

3. Écrire la procédure `afficherFraction` qui affiche une fraction sous la forme `n/d`. Si le numérateur est plus grand que le dénominateur (en valeur absolue), on affichera la fraction sous la forme `a + m/d` où `a` est un entier et `n/d = a +m/d`.

4. Écrire la méthode `formeIrreductible` qui permet de transformer une fraction en forme irréductible. Une fraction irréductible est une fraction `a/b` où le numérateur `a` est un entier et le dénominateur `b` un entier positif, de telle façon qu'il n'existe pas d'autre fraction `c/d` représentant le même nombre avec `c` plus petit que `a` en valeur absolue, `c` et `d` étant également deux entiers. Si une fraction `n/d` n'est pas irréductible, alors elle peut être réduite, c'est-à-dire que `n` et `d` peuvent être divisés par un même nombre, le plus grand possible. Ce nombre est le `PGCD` de `n` et `d`.

5. Écrire les différentes opérations binaires (`addition`, `soustraction`, `multiplication` et `division`) pour les fractions. Elles prendront en paramètre 2 fractions et retourneront le résultat sous forme de fraction irréductible.

Rappel :
PGCD(a,b) = PGCD(a−b,b) ou PGCD(a,b) = PGCD(b, r) avec r le reste de la division entière de a par b.

Exercice : Statistiques (1 + 1 + 2 = 4)

Nous souhaitons faire un peu de statistiques sur les notes des étudiants à l'EISTI. Les notes seront stockées dans un tableau, la taille et les valeurs seront saisies par l'utilisateur.

1. Écrire la fonction `saisirNotes` qui prend en paramètre la taille du tableau, qui demande les notes à l'utilisateur et qui renvoie le tableau de notes.
2. Écrire la fonction `moyenne` qui calcul et renvoie la moyenne des notes stockées dans le tableau.
3. Écrire la fonction `variance` qui calcul et renvoie la variance des notes stockées dans le tableau.

Rappel :

La formule de la variance σ^2 est donnée en fonction de la moyenne $\overline{x}$ par : $\sigma^2 = \frac{\sum(x-\overline{x})^2}{n}$