

INFO-H-100 : Introduction à la programmation

Projet 2 : *Des chiffres et des lettres*

On vous demande de réaliser un jeu “*Des chiffres et des lettres*”¹ simplifié en C++. Il s’agit d’un jeu télévisé reposant sur les compétences en calcul mental et sur la connaissance du vocabulaire des candidats.

La section suivante est basée sur Wikipedia et sur le règlement officiel du jeu².

Déroulement du jeu

Le jeu oppose deux candidats et est basé sur deux types d’épreuves : *Le compte est bon* et *Le mot le plus long*. Un match se joue en deux manches qui se composent de 4 coups de chiffres et de 8 coups de lettres. Une manche commence par deux coups de lettres suivis d’un coup de chiffres et ainsi de suite.

Le candidat ayant marqué le plus de points remporte la manche. En cas d’égalité à la fin d’une manche, les candidats seront départagés par un nouveau *Le mot le plus long*. Si les candidats gagnent chacun une manche, ils devront jouer une troisième manche. Les compteurs de points sont remis à zéro après chaque manche.

Le mot le plus long

Le but est de trouver, en 30 secondes, le mot le plus long possible à l’aide d’un ensemble de 9 lettres.

Chaque candidat demande alternativement soit une consonne soit une voyelle tirée aléatoirement parmi un ensemble de 200 consonnes et 200 voyelles. Chaque tirage doit être composé au minimum de 2 voyelles. Les joueurs ont ensuite 30 secondes pour trouver, à l’aide des 9 lettres tirées, le mot le plus long existant dans un dictionnaire donné.

On attribue 1 point par lettre utilisée. Si un candidat propose un mot qui n’existe pas, le nombre de points ne lui sera pas attribué et c’est l’autre candidat qui recevra ces points.

Le compte est bon

Il s’agit pour les deux candidats d’effectuer, en 45 secondes, un certain nombre d’opérations d’arithmétique élémentaire (addition, division, soustraction ou multiplication) afin de retrouver un nombre de 3 chiffres choisi aléatoirement ou de s’en approcher le plus possible.

Pour ce faire, 6 plaques sont tirées au sort parmi un ensemble composé de deux séries de 10 plaques numérotées de 1 à 10 et une série quatre plaques portant les nombres 25, 50, 75 et 100, soit un ensemble de 24 plaques. Dans les calculs effectués par les candidats, chacune des 6 plaques ne peut être utilisée qu’une seule fois mais les candidats ne sont pas tenus d’utiliser toutes les plaques.

Si un joueur trouve le bon compte, il gagne 9 points. Sinon, le joueur qui s’est le plus approché du bon compte gagne 6 points. Il peut bien sûr y avoir égalité.

¹http://fr.wikipedia.org/wiki/Des_chiffres_et_des_lettres

²<http://cybercl.free.fr/reglem/freglem.html>

Programme à réaliser

On vous demande de réaliser à l'aide uniquement de la matière vue au cours et aux séances d'exercices un jeu de *Des chiffres et des lettres simplifié à un seul joueur* en C++. Votre programme devra se dérouler comme suit :

- Tout d'abord, votre programme chargera en mémoire le dictionnaire ainsi que l'ensemble des lettres disponibles sous forme de fichiers sur la page web des projets³. Ces fichiers devront obligatoirement se trouver dans le même répertoire que l'exécutable de votre projet. Le format de ces fichiers est détaillé ci-dessous.
- Ensuite, le jeu peut commencer. Un jeu se compose de 4 coups de chiffres (*Le compte est bon*) et de 8 coups de lettres (*Le mot le plus long*). Il commence par deux coups de lettres suivis d'un coup de chiffres et ainsi de suite.

Le mot le plus long

Un coup de lettres se déroule de la manière suivante :

- Le candidat demande à l'ordinateur de tirer aléatoirement une voyelle ou une consonne parmi celles disponibles jusqu'à atteindre un ensemble de 9 lettres composé d'au moins deux voyelles. Ces lettres sont retirées jusqu'à la fin du jeu.
 - Le candidat propose un mot à l'aide de ces 9 lettres. Le système vérifie alors les conditions suivantes :
 - Le mot existe dans le dictionnaire fourni. La recherche dans le dictionnaire sera faite de manière efficace. Les diacritiques sont ignorés.
 - Le mot est composé uniquement des lettres à la disposition du joueur. Chaque lettre ne peut être utilisée qu'une seule fois.
- Si ces conditions sont vérifiées, le candidat gagne autant de points que le nombre de lettres constituant le mot proposé. Sinon, il perd ce nombre de points.

Par exemple, si le tirage est LORICPISE, on peut jouer POLICIERS.

Le compte est bon

Un coup de chiffres se déroule de la manière suivante :

- Le programme tire aléatoirement 6 plaques dans l'ensemble de plaques disponibles et invente un nombre entre 100 et 999.
- Le candidat propose le nombre qu'il a trouvé à l'aide de ces 6 plaques et des opérateurs de base (+, -, *, et /) et propose un calcul. Chaque plaque ne peut être utilisée qu'une seule fois mais toutes les plaques ne doivent pas être utilisées.
- Si le candidat a trouvé le nombre exact, il gagne 9 points. Si la différence entre le nombre à trouver et le nombre trouvé est inférieur à 9, il gagne 9 moins cette différence. Sinon il ne gagne aucun point.
- A la fin du coup, les plaques sont remises dans le pot.

Voici un exemple d'exécution de cette partie du jeu :

```
Disponibilités : 3 75 2 4 1 1
But            : 888
Proposition ?  : 888
Plaque ?      : 75
Opérateur ?   : -
Plaque ?      : 1
Disponibilités : 3 2 4 1 74
Plaque ?      : 3
Opérateur ?   : *
Plaque ?      : 4
Disponibilités : 2 1 74 12
Plaque ?      : 74
Opérateur ?   : *
Plaque ?      : 12
```

³<http://cs.ulb.ac.be/public/teaching/infoh100/projets>

Bravo, vous avez gagné 9 points

Structure des fichiers

Le fichier `dico.txt` contient une liste de mots triés alphabétiquement. Les mots sont séparés par un retour à la ligne.

Dans les fichiers `voyelles.txt` et `consonnes.txt`, chaque ligne est composée d'une lettre et du nombre d'occurrences de cette lettre dans le jeu séparés par un espace. Extrait du fichier `voyelles.txt` :

```
E 40  
A 66  
I 36
```

Apports personnels

Les apports personnels seront valorisés à hauteur de 5 points sur 20. En d'autres mots, un projet parfait sans apport personnel aura une valeur de 15 points sur 20. Ces apports personnels peuvent être par exemple des règles ignorées dans la version simplifiée proposée ci-dessus. Voici quelques exemples d'apports personnels :

- Gestion de deux joueurs (difficulté moyenne) ;
- Enregistrer une partie sur fichier et pouvoir la rejouer (difficile) ;
- Algorithmes de recherche de la meilleure solution (très difficile) ;
- Affichage amélioré grâce à l'Art ASCII⁴ (difficulté moyenne) ;
- Enregistrement des meilleurs scores (facile) ;
- Gestion du temps. Lors d'un coup de lettres, le joueur a par exemple 30 secondes pour valider sa réponse. Lors d'un coup de chiffres, le joueur a 45 secondes pour proposer un nombre et une minute pour encoder son raisonnement. (difficulté moyenne) ;
- Fichier de configuration (temps, difficulté, nombre de lettres d'un mot, ...) (difficulté moyenne).

⁴http://fr.wikipedia.org/wiki/Art_ASCII

Rapport

Un rapport dactylographié vous est demandé. Celui-ci devra comprendre :

- Une page de garde indiquant clairement vos noms, numéros de matricules et votre série.
- Une **brève** introduction au problème.
- Une documentation de vos fonctions, c'est à dire pour chaque fonction : son prototype, une description succincte de la tâche qu'elle remplit, ses paramètres et leur domaine (le type et les valeurs possibles), le type de la valeur de retour et son domaine, les erreurs possibles et un exemple d'utilisation. Par exemple :

```
int daysInMonth(int month, int year) : Calcule le nombre de jours du mois (month) de l'année (year) fourni en paramètres.
```

 - Paramètre `int month` : entier de 1 à 12 représentant le mois.
 - Paramètre `int year` : entier de 0 à 9999 représentant l'année au format AAAA.
 - Valeur de retour entière indiquant le nombre de jours du mois.
 - Erreurs : renvoie -1 si les paramètres ne respectent pas les domaines.
 - Exemple : `daysInMonth(04,2008)` renvoie l'entier 30.
- Une explication claire de vos choix algorithmiques pour les fonctions qui le nécessitent.
- Un manuel d'utilisation illustré de votre programme fini de maximum 2 pages.
- Le code source en police de caractère à empattement fixe (par exemple `courier`).

Consignes

Le projet se fera obligatoirement par groupes de deux étudiants de la même série.

Le rapport au format papier (uniquement des feuilles et des agraffes, **pas de plastique** ni de reliure) devra être rendu au bureau UB4.131. Le code source de votre application sera envoyé par mail à l'adresse `ulb.code+infoh100@gmail.com`. Ce mail devra avoir comme sujet "INFO-H-100 - Projet 2 - Nom1, Prénom1 et Nom2, Prénom2". Tout cela devra être rendu pour le **30 avril à 12h30**. Les groupes remettant leur projet après cette limite seront pénalisés.

Vous devrez défendre votre projet devant les assistants à un horaire fixé ultérieurement.

Votre code devra compiler sans avertissement à l'aide de la commande `g++ -Wall`, version 4.0.3 installée dans les salles machines. Les seules bibliothèques autorisées sont `iostream`, `ctime`, `cstdlib` et `fstream`.

L'évaluation de ce projet prendra en compte les points suivants :

- La résolution du problème énoncé.
- Le respect scrupuleux de l'énoncé.
- Le respect scrupuleux des conventions et des règles de bonne pratique publiés sur le site web des projets.
- La qualité du rapport : intérêt, présentation, soin, concision, grammaire et orthographe.
- Votre défense en salle machine devant les assistants.
- L'efficacité de vos algorithmes et structures de données, en restant bien sûr dans le cadre de la matière vue aux cours et aux séances d'exercices.

Les élèves assistants sont disponibles tous les midis dans la salle Socrate afin de répondre à vos questions.

Bon travail!

Lecture et écriture sur fichier

La librairie `fstream` est une librairie standard permettant de lire et écrire dans un fichier. Cette librairie offre deux nouveaux types de données : `ifstream` pour la lecture de fichier et `ofstream` pour l'écriture. Leur fonctionnement est similaire à celui de `cin` et `cout`.

Voici un exemple de lecture d'un fichier :

```
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    char chaine[255];
    ifstream entree("test.txt");    //ouvre le fichier test.txt en lecture
                                    //ce fichier se trouve dans le meme dossier
                                    //que le programme

    entree >> chaine;
    while (!entree.eof())          //tant que le fichier n'est pas terminé
    {

        cout << chaine << endl;
        entree >> chaine;          //copie une portion dans le vecteur chaine
                                    //(chaque portion est séparée par un espace,
                                    // une tabulation ou un retour à la ligne)

        entree.close();            //ferme le fichier

    }

    return 0;
}
```

Voici un exemple d'écriture d'un fichier :

```
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    int valeur;
    ofstream sortie("truc.txt");    //ouvre le fichier truc.txt en écriture

    cin >> valeur;
    while (valeur != -1)
    {
        sortie << valeur << ',' ;    //écrit dans les fichiers les valeurs lues
        cin >> valeur;                //au clavier
    }

    sortie.close();                //ferme le fichier

    return 0;
}
```