

### 1 Jeu du pendu

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 mauvaises propositions 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 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 renvoie un mot pris aléatoirement dans le fichier.
- Remarque : Un fichier "mots.txt" est fourni, contenant une liste de mots prédéfinis.* ☐

- ② Écrire la procédure `void jouePendu (char* motATrouver)` qui prend en paramètre le mot à faire deviner et qui déroule le jeu. Les différentes étapes du jeu sont :
1. Faire une copie du mot et remplacer tous les caractères de la copie par un caractère marquant la non-découverte de la lettre;
  2. Tant que tout le mot n'a pas été découvert et qu'il reste encore des essais
    - (a) Demander une lettre
    - (b) Jouer cette lettre
  3. Annoncer si le joueur a gagné (en combien de coups) ou s'il a perdu.
- ☐

- ③ 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`. ☐
  
- ④ On souhaite garder les scores (nombre d'essai restant) des joueurs.  
À la fin de la partie, demander le pseudo du joueur, et sauvegarder le score dans un fichier. Si le pseudo existe déjà dans le fichier de score, on garde le meilleur score. ☐