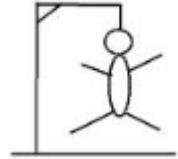


TP 8 : le jeu du pendu



a) Écrire une fonction qui range les mots d'un fichier de mots (un par ligne) dans un tableau de `String`. Comme on ne connaît pas à l'avance le nombre de mots du fichier, on crée un tableau de 100.000 cases. Ce tableau ne sera jamais rempli complètement. On mémorise dans la case 0 le nombre de mots effectivement stockés et les mots commencent à l'indice 1.

b) Écrire une fonction qui, étant donné le tableau précédent, retourne un mot au hasard. Testez vos deux fonctions avec un *main* qui appelle ces deux fonctions et affiche le mot choisi. Vous pouvez utiliser le fichier de mots `dico8095.txt` (disponible sur Moodle) qui contient 8095 mots fréquents du français sans lettres accentuées et d'au moins 3 lettres.

c) Compléter le *main* pour programmer le jeu du pendu : l'ordinateur choisit un mot au hasard que l'utilisateur essaie de deviner lettre par lettre. Au départ, l'ordinateur présente un gabarit : la première lettre, la dernière lettre ainsi qu'un tiret à la place de chacune des autres lettres. Par exemple :

O - - - - - **R**

L'utilisateur propose des lettres l'une après l'autre. Chaque fois qu'une lettre est présente dans le mot, toutes ses occurrences sont affichées. Dans le cas contraire, l'ordinateur ne modifie rien. Le jeu se poursuit tant qu'il y a des lettres non trouvées (voir l'exemple ci-contre).

Commencez par écrire quelques lignes de code pour construire, à partir de la chaîne du mot, le gabarit initial (la première lettre du mot, des tirets et la dernière lettre du mot). Ensuite, au sein d'une grande boucle, écrire les instructions permettant de demander une lettre à l'utilisateur et de passer en revue chacune des lettres du mot pour reconstruire le nouveau gabarit, soit en y recopiant la lettre correspondante du gabarit, soit en plaçant la lettre trouvée par l'utilisateur. Puis le nouveau gabarit est affecté au gabarit courant.

d) Modifiez le jeu en ajoutant un compteur. En plus du mot, l'ordinateur indique le nombre de coups déjà joués. Dès que l'on arrive à 10, le jeu est terminé. Exemple :

T - - - - - E

Lettre ? **V**

T - - - - - E (1)

Lettre ? **T**

T - - T - - - T T E (2)

Lettre ? **O**

...

Lettre ? **R**

T - R T - - - T T E (10)

Vous avez perdu.

e) **Bonus** : En fin de partie, affichez un score qui est le nombre d'essais moins le nombre de lettres différentes à deviner. Modifiez le programme pour que l'on puisse rejouer après une partie. L'ordinateur devra alors mémoriser le meilleur score (le plus petit) et indiquer à l'utilisateur à la fin de chaque partie s'il a le meilleur score ou non.

f) **Bonus 2** : Stockez les 10 meilleurs scores et les noms des joueurs correspondant dans un fichier texte qui sera mis à jour quand un joueur battra le 10^e meilleur score.