

DEVOIR 3

INFO 1101 – Principes de programmation I

À remettre le jeudi 14 avril 2022 à 23 h 59. Aucun retard ne sera accepté.

Ce devoir vise à implémenter trois fonctions pour réaliser un clone complet du jeu [Wordle](#).

Préparation

Téléchargez les fichiers `devoir3.cpp` et `devoir3-test.h` sur [Clic](#). Ne modifiez pas le fichier `devoir3-test.h`. Le fichier `devoir3.cpp` est à compléter en suivant les étapes ci-dessous. **Assurez-vous de suivre les instructions données dans le fichier `instructions.pptx`!**

Le jeu

Le jeu consiste à deviner un mot caché de 5 lettres (la solution) en utilisant un maximum de 6 essais. Chaque essai doit être un mot de 5 lettres figurant dans le dictionnaire (fourni). Chaque essai est comparé à la solution lettre par lettre, puis les résultats sont affichés ainsi :

- Les lettres **correctes** sont colorées en vert. Les lettres correctes de l'essai figurent dans la solution au même endroit :

solution :	A	B	C	E	S
essai :	A	B	I	M	E

- Les lettres **présentes** sont colorées en jaune. Les lettres présentes de l'essai figurent dans la solution, mais à un autre endroit :

solution :	A	B	C	E	S
essai :	A	B	I	M	E

- Les lettres **absentes** sont colorées en gris foncé. Les lettres absentes ne sont pas présentes dans la solution :

solution :	A	B	C	E	S
essai :	A	B	I	M	E

Étapes à suivre

Le devoir s'effectue en trois parties, qui correspondent à chacune des trois fonctions à implémenter.

Partie 1 : Choisir une solution au hasard

Créez une fonction appelée `choisirIndiceSolutionAuHasard`. La fonction doit retourner une valeur de type `int` et elle doit accepter un paramètre de type `int` représentant le nombre de solutions. Assurez-vous d'inclure le prototype de la fonction.

La fonction doit retourner un nombre **choisi au hasard** dans l'intervalle $[0, N - 1]$, où N représente le nombre de solutions reçu en paramètre.

Vous pouvez retourner une valeur fixe pendant que vous développez. Par exemple, si vous retournez toujours la valeur zéro, le premier mot du dictionnaire, `ABATS`, sera toujours choisi comme solution. Assurez-vous de retourner une valeur aléatoire avant de remettre votre devoir.

Partie 2 : Mettre un mot en majuscules

Créez une fonction appelée `mettreMotEnMajuscules`. La fonction ne doit rien retourner (`void`) et elle doit accepter comme paramètre un tableau de caractères représentant un mot (il peut s'agir d'une réponse de l'utilisatrice ou de l'utilisateur, ou bien de la solution). Assurez-vous d'inclure le prototype de la fonction.

Le mot reçu en paramètre aura toujours au moins 5 lettres, donc le tableau contiendra toujours au moins 5 éléments. Le mot peut ou non se terminer par le caractère zéro (`\0`), mais vous n'avez pas à le traiter. Lorsque vous parcourez le tableau, vous devez utiliser la constante `LONGUEUR_MOT` définie dans le fichier `devoir3-test.h` plutôt que d'écrire le chiffre 5.

Votre fonction doit remplacer chaque lettre minuscule du mot reçu en paramètre par son équivalent en majuscules. Toutefois, le mot pourrait déjà être en majuscules, donc vous devez vous assurer de ne pas modifier les lettres qui sont déjà en majuscules.

Partie 3 : Évaluer une réponse

Créez une fonction appelée `evaluerReponse`. La fonction ne doit rien retourner (`void`) et elle doit accepter les trois paramètres suivants :

- 1) Un tableau de caractères représentant la **réponse** entrée par l'utilisatrice ou l'utilisateur. Ce tableau contiendra au moins 5 lettres en majuscules.
- 2) Un tableau de caractères représentant la **solution** choisie au hasard. Ce tableau contiendra au moins 5 lettres en majuscules.
- 3) Un tableau qui contient 5 éléments de type `ResultatLettre` (un type énuméré défini dans le fichier `devoir3-test.h`, à la ligne 1323). Les 5 éléments correspondent aux **résultats** des 5 lettres de la réponse fournie en paramètre.

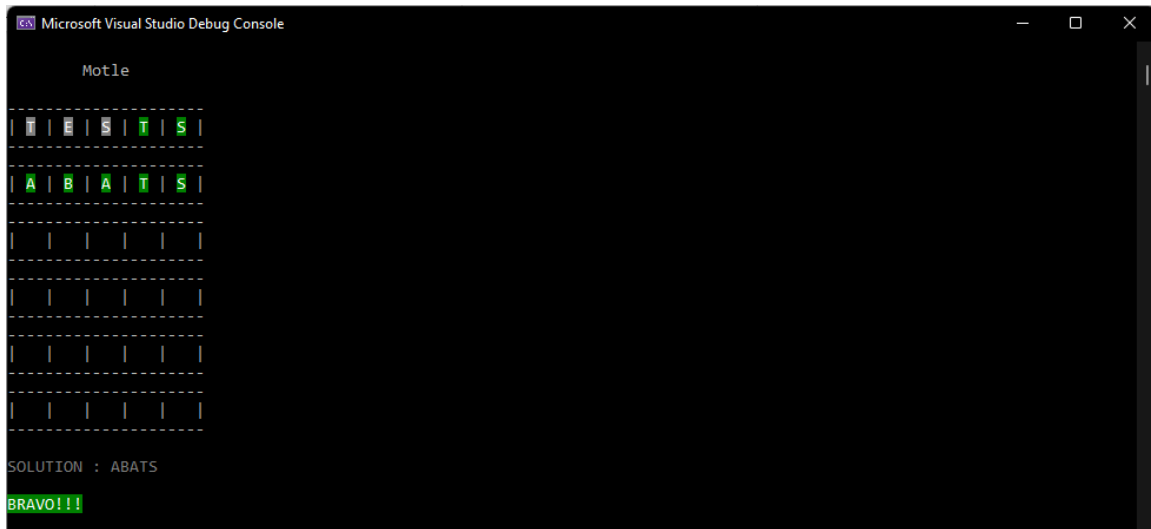
Votre fonction doit comparer la réponse et la solution fournies en paramètres et modifier le tableau de résultats selon la méthode vue en classe (qui est aussi décrite dans le fichier **devoir3.pptx** disponible sur Clic). Chaque fois que votre fonction est appelée, le tableau de résultats est initialisé au préalable pour contenir dans chacune des cases la valeur `ResultatLettre::ABSENTE`. Assurez-vous d'inclure le prototype de la fonction.

Remarque

Ne modifiez pas la fonction `main()`, sauf aux deux endroits indiqués par des commentaires :

- 1) Vous pouvez modifier la langue afin d'utiliser le dictionnaire original en anglais.
- 2) Vous pouvez aussi modifier la valeur booléenne du cinquième paramètre de la fonction `jouer()` pour faciliter le développement. Si vous mettez cette valeur à `true`, la solution choisie au hasard sera affichée, et l'utilisatrice ou l'utilisateur pourra entrer n'importe quel mot de 5 lettres, qu'il soit ou non dans le dictionnaire. Si vous mettez cette valeur à `false`, la solution sera cachée et les véritables règles du jeu s'appliqueront.

Exemple de sortie



```
Microsoft Visual Studio Debug Console

Motle

| T | E | S | T | S |
|---|
| A | B | A | T | S |
|---|
|   |   |   |   |   |
|---|
|   |   |   |   |   |
|---|
|   |   |   |   |   |
|---|
|   |   |   |   |   |
|---|

SOLUTION : ABATS
BRAVO!!!
```

Veuillez remettre uniquement le fichier `devoir3.cpp` sur Clic. N'incluez pas les exécutables ni aucun autre fichier. Ne remettez pas non plus le fichier `devoir3-test.h` fourni.