Large Program 3
Hangman

COP3223C Introduction to Programming with C Fall 2022
Dr. Andrew Steinberg

Background Story of this Assignment

You are about to implement your third large program in this course. In this assignment, the primary objective is to apply your skills in strings!! Strings are very important to understand and this assignment will explore some of the topics discussed from our lectures.

You will write a menu driven program in C that simulates the classic word game Hangman. For those that never played hangman, click here to see the rules. In this large programming assignment, you will write a simple terminal hangman game application.

The program will have the following setup.

- The program will first welcome the user with a friendly message. It will provide the user with instructions and rules on how the game works.
- After the welcome message, the game will begin with the first round.
- For each round:
  - User will be asked to enter a letter and will a show masked word with *'s.
  - Program will inform user if letter is in the word.
- If the letter is in the word, the program will reveal where in the masked word the letter occurs.
- If the letter is not in the word, the program will inform the user that a strike has been received.
- After the round, the program will determine if user won or lost.
- Program will ask user if the user wants to play again.

In order to create this program, you will use the following function prototypes that are provided for you in the assignment directions.

Requirements for Large Program 3

Requirements for the Program:

1. The program will only deal with single words (NO PHRASES!).

2. The words must be read from a text file. The text file is provided for you on Webcourses. The text file must be in the same directory as your C source file. Note: If you are first working locally on your machine and not Eustis, you will have to provide a file path. If you go this route, you are responsible for testing on Eustis which means you will have to remove the file path so it doesn't crash when your grader tests it!

3. The max number strikes you can go up to are 6. Use a macro constant!

4. The max size of words used can only go up to 20 characters. Use a macro constant!

5. Words can have the same letter in different places. *For example, state, cellphone, and paper are words that can be used in the game.*

6. Program must be able to properly handle any kind of input without crashing.

Due Date

The assignment is due on November 22nd at 11:59pm EST via Webcourses. **Do not email the professor or TAs your submissions as they will not be accepted!** This assignment is accepted late up to 24 hours with a penalty. Please see the syllabus for more information on this. Make sure to submit on time to get potential full credit. Make sure to also take into consideration the uploading time. In the past, students who are working last minute on the assignment sometimes run into uploading issues where their Internet may run slow, resulting in late submissions. The timestamp Webcourses uses for your submission will be applied and will be the final say. Please do not email the instructor or TAs saying your Internet was running slow. If the time is off by a second of the due date, then the assignment is considered late. Plan accordingly!

Important! Read Carefully!

In this assignment, you are going to implement **6 user defined functions**

```
void rules(void); //display rules of the game
void maskWord (char starword[], int size); //mask the word with stars to display
int playRound(char starword[], char answer[]); //play a round of hangman
int occurancesInWord(char userguess, char answer[]); //number of times letter occurs in word
void updateStarWord(char starword[], char answer[], char userguess); //replace respective *
void playAgain(int *play); //ask user if to play again. 1 is yes 2 is no
```

You cannot add any additional functions or remove them. Modifying them in any way (name, parameters, etc...) will result in point deductions! Utilize them the way they are provided!

The function prototypes are provided for you here. You will create a file called `largeprogram3_lastnamefirstname.c` for this assignment and write out the code. There is NO

SKELETON PROVIDED FOR THIS LARGE PROGRAM! **Do NOT modify any of the function prototypes that are provided for you!** Any modifications from the provided function prototypes will result in points being deducted! You are allowed to use the same message that is provided in the sample output for your program. You can also make modifications to the text of messages as long as the program follows all directions as stated in this assignment.

The following section will discuss the prototypes more in detail. Make sure that you name your C file *largeprogram3_lastname_firstname.c*, where lastname and firstname is your last and first name (as registered in webcourses). For example Dr. Steinberg's file would be named *largeprogram3_Steinberg_Andrew.c*. Make sure to include the underscore character _. If your file is not named properly, points will be deducted. You are also provided a text file called 'words.txt' containing words to use for the game.

The Function Prototypes

This section will discuss the function prototypes.

```
void rules(void); //display rules of the game
```

This function will display the rules of the game. See the sample output to get a general idea of what is displayed to the user when the program runs.

```
void maskWord (char starword[], int size); //mask the word with stars to display
```

The mask function will "mask" the solution word with the * character for the user to see on the terminal window. This will allow the user to keep track of letters that were guessed correctly in forming the word solution. The function has two parameters. The first parameter is a string that will represent the starword. The second parameter is an integer that represents the number of characters for a particular solution in the round of hangman.

```
int playRound(char starword[], char answer[]); //play a round of hangman
```

The playRound function simulates an entire round of the Hangman game. The function returns an integer representing the outcome of the game. If 1 is returned, then the user won. Otherwise return 0 if the user lost. The function has two parameters. The first parameter is a string to the starword (string mixed with *'s and letters) and the second parameter is another string representing the solution. This function serves as the heart of the game. This function will have to call other functions except for `playAgain`, `maskWord`, and `rules`. Think carefully how this would work.

```
int occurancesInWord(char userguess, char answer[]); //number of times letter occurs in word
```

The occurancesInWord function counts the number of times a letter occurs in the solution. If the function returns a positive number, then that means it occurs at least once. Otherwise return 0 if it doesn't occur at all in the solution. This function has two parameters. The first parameter is a character that represents the letter guessed by the user. The second parameter is the answer string.

```
void updateStarWord(char starword[], char answer[], char userguess); //replace respective *
```

The updateStarWord function will update the masked string by replacing the respective * character(s) with the the corresponding the letter that was guessed correctly. The function has three parameters. The first parameter is a string to the masked word. The second parameter is another string representing the answer. The third parameter is the letter that the user guessed.

```
void playAgain(int *play); //ask user if to play again. 1 is yes 2 is no
```

The playAgain function will ask the user if they would like to play another round of hangman. The user will input an option and it will be stored in the play variable. The function has one parameter which represents an integer reference to the variable that keeps track if the user will play again or not.

Testing on Eustis

It is your responsibility to test your code on Eustis. If you submit your assignment without testing on Eustis, you risk points being deducted for things that may behave differently on your operating system. Remember, you cannot dispute grades if your code didn't work properly on Eustis all because it worked on your machine. The Eustis environment gives the final say. Plan accordingly to test on Eustis!!

Why was I not provided a Python Script File?

You are probably wondering why Dr. Steinberg didn't provide a python script to check your code like in the small programs. The reason is that Dr. Steinberg wants his students to enjoy the assignment without being told of what must be displayed to the terminal and matching exactly. For large programs, Dr. Steinberg allows students to change the text output to the terminal, however your program must perform the specific functionalities that is requested at least to receive potential full credit.

The Rubric

Please see the assignment page for the established rubric on webcourses.

Comment Header

Make sure you place a comment header at the top of your C file. You will use single line comments to write your name, professor, course, and assignment. For example, Dr. Steinberg's header would be:

```
//Andrew Steinberg
//Dr. Steinberg
```

```
//COP3223C Section 1
//Large Program 3
```

Missing a comment header will result in point deductions!

The Solution Text File

I have provided a sample output of a working program of what is expected. As I stated before, you are welcome to change the text output to the terminal, or you can keep it exact like in the sample.

Large Program 3 Sample Text File Solution

```
Welcome to the Hangman Game!
Here are the rules.
I will provide you a set of Each You must figure out each letter of the missing
↪  word.
For every correct letter guessed, I will reveal its place in the word.
Each mistake will result in a strike.
6 strikes will result in a loss that round.
Are you ready? Here we go!
Welcome to the Round!
The size of the word has 4 letters.

You currently have 0 strikes.
Letters you have guessed:


Enter your guess: A

The letter a is NOT in the word.

You currently have 1 strikes.
Letters you have guessed: a


Enter your guess: G

The letter g is in the word.

You currently have 1 strikes.
Letters you have guessed: ag


Enter your guess: f

The letter f is in the word.

You currently have 1 strikes.
Letters you have guessed: agf
```

```
f
Enter your guess: 0

You did not enter a letter from the alphabet.

You currently have 1 strikes.
Letters you have guessed: agf

f
Enter your guess: !

You did not enter a letter from the alphabet.

You currently have 1 strikes.
Letters you have guessed: agf

f
Enter your guess: o

The letter o is in the word.

You currently have 1 strikes.
Letters you have guessed: agfo

f
Enter your guess: r

Congratulations! You won! The word was frog.
Would you like to play another round?
1: Yes
2: No
Choice: 1

Welcome to the Round!
The size of the word has 6 letters.

You currently have 0 strikes.
Letters you have guessed:


Enter your guess: a

The letter a is NOT in the word.

You currently have 1 strikes.
Letters you have guessed: a


Enter your guess: e
```

```
The letter e is in the word.

You currently have 1 strikes.
Letters you have guessed: ae


Enter your guess: i

The letter i is in the word.

You currently have 1 strikes.
Letters you have guessed: aei


Enter your guess: o

The letter o is NOT in the word.

You currently have 2 strikes.
Letters you have guessed: aeio


Enter your guess: u

The letter u is NOT in the word.

You currently have 3 strikes.
Letters you have guessed: aeiou


Enter your guess: q

The letter q is NOT in the word.

You currently have 4 strikes.
Letters you have guessed: aeiouq


Enter your guess: w

The letter w is in the word.

You currently have 4 strikes.
Letters you have guessed: aeiouqw

wi
Enter your guess: r

The letter r is in the word.

You currently have 4 strikes.
```

```
Letters you have guessed: aeiouqwr

wi
Enter your guess: t

The letter t is in the word.

You currently have 4 strikes.
Letters you have guessed: aeiouqwrt

wi
Enter your guess: y

The letter y is NOT in the word.

You currently have 5 strikes.
Letters you have guessed: aeiouqwrty

wi
Enter your guess: p

The letter p is NOT in the word.
Sorry you did not win the round. The word was winter.
Would you like to play another round?
1: Yes
2: No
Choice: 1

Welcome to the Round!
The size of the word has 9 letters.

You currently have 0 strikes.
Letters you have guessed:


Enter your guess: a

The letter a is in the word.

You currently have 0 strikes.
Letters you have guessed: a


Enter your guess: e

The letter e is in the word.

You currently have 0 strikes.
Letters you have guessed: ae
```

```
Enter your guess: i

The letter i is in the word.

You currently have 0 strikes.
Letters you have guessed: aei


Enter your guess: o

The letter o is NOT in the word.

You currently have 1 strikes.
Letters you have guessed: aeio


Enter your guess: u

The letter u is NOT in the word.

You currently have 2 strikes.
Letters you have guessed: aeiou


Enter your guess: s

The letter s is in the word.

You currently have 2 strikes.
Letters you have guessed: aeious

s
Enter your guess: b

The letter b is NOT in the word.

You currently have 3 strikes.
Letters you have guessed: aeiousb

s
Enter your guess: c

The letter c is in the word.

You currently have 3 strikes.
Letters you have guessed: aeiousbc

s
Enter your guess: T

The letter t is in the word.
```

```
You currently have 3 strikes.
Letters you have guessed: aeiousbct

stai
Enter your guess: 5

You did not enter a letter from the alphabet.

You currently have 3 strikes.
Letters you have guessed: aeiousbct

stai
Enter your guess: r

Congratulations! You won! The word was staircase.
Would you like to play another round?
1: Yes
2: No
Choice: 2

Thank you for playing today!
```

Recommendations for Completing the Assignment

At this point in the course, you should start to build a foundation of building programs without Dr. Steinberg directly telling you where to begin. Something Dr. Steinberg will point out is that you can use the sample output to help you build your program, HOWEVER, make sure to consider and think carefully where those lines of output are occurring (i.e. main function or one of the user defined functions). Please don't feel discourage that Dr. Steinberg didn't provide this as it wasn't on purpose to make the course harder (it was with good intentions). We (Dr. Steinberg and his TAs/ULAs) are still here to help you succeed! My objective is to help you build your programming and problem solving skills for your future foundational CS courses such as CS1. If you having trouble of where to begin or where something should occur, please come see us!!! We will still provide the same level of help!

Something Dr. Steinberg will provide to assist is how the main function should work in this assignment to help you get started. Inside the main function you are going to welcome to the user to the game. After welcoming the user, you will then have to open the word text file (provided in Webcourses) and read one string at a time. Once a string is read successfully, create the masked version (hint hint `maskWord`) and then begin the actual round (hint hint call `playRound`)! Once `playRound` ends you will then ask the user if they want to play again (OMG this sounds like the function `playAgain` right!?!) Based on the result, you either have to repeat the action (scan new word and play a new round, or terminate the program).

Some Advice

Here are some tips and tricks that will help you with this assignment and make the experience enjoyable.

- Do not try to write out all the code and build it at the end to find syntax errors. For each new line of code written (my rule of thumb is 2-3 lines), build it to see if it compiles successfully. **It will go a long way!**
- After any successful build, run the code to see what happens and what current state you are at with the program writing so you know what to do next! If the program performs what you expected, you can then move onto the next step of the code writing. If you try to write everything at once and build it successfully to find out it doesn't work properly, you will get frustrated trying find out the logical error in your code! **Remember, logical errors are the hardest to fix and identify in a program!**
- Start the assignment early! Do not wait last minute (the day of) to begin the assignment.
- Ask questions! It's ok to ask questions. If there are any clarifications needed, please ask TAs and the Instructor! We are here to help!!! You can also utilize the discussion board on Webcourses to share a general question about the large program as long as it doesn't violate the academic dishonesty policy.