ASSIGNMENT 5

```c
#include <ctype.h>

#include <stdbool.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#define MAX_WORD_LENGTH 50

#define MAX_TRIES 6

struct WordWithHint {

        char word[MAX_WORD_LENGTH];

        char hint[MAX_WORD_LENGTH];

};

void displayWord(const char word[], const bool guessed[]);

void drawHangman(int tries);

int main()

{

        srand(time(NULL));

        struct WordWithHint wordList[] = {

                { "geeksforgeeks", "Computer coding" },

                { "elephant", "A large mammal with a trunk" },

                { "pizza", "A popular Italian dish" },

                { "beach", "Sandy shore by the sea" },

        };

        int wordIndex = rand() % 4;

        const char* secretWord = wordList[wordIndex].word;

        const char* hint = wordList[wordIndex].hint;
```

```c
    int wordLength = strlen(secretWord);

    char guessedWord[MAX_WORD_LENGTH] = { 0 };

    bool guessedLetters[26] = { false };

    printf("Welcome to Hangman!\n");

    printf("Hint: %s\n", hint);

    int tries = 0;

    while (tries < MAX_TRIES) {

            printf("\n");

            displayWord(guessedWord, guessedLetters);

            drawHangman(tries);

            char guess;

            printf("Enter a letter: ");

            scanf(" %c", &guess);

            guess = tolower(guess);

            if (guessedLetters[guess - 'a']) {

                    printf("You've already guessed that letter. "

                            "Try again.\n");

                    continue;

            }

            guessedLetters[guess - 'a'] = true;

            bool found = false;

            for (int i = 0; i < wordLength; i++) {

                    if (secretWord[i] == guess) {

                            found = true;

                            guessedWord[i] = guess;

                    }

            }

            if (found) {

                    printf("Good guess!\n");

            }

            else {
```

```c
                    printf("Sorry, the letter '%c' is not in the "
                            "word.\n",
                            guess);

                    tries++;
            }

            if (strcmp(secretWord, guessedWord) == 0) {
                    printf("\nCongratulations! You've guessed the "
                            "word: %s\n",
                            secretWord);

                    break;
            }
    }
    if (tries >= MAX_TRIES) {
            printf("\nSorry, you've run out of tries. The word "
                    "was: %s\n",
                    secretWord);
    }


    return 0;
}
void displayWord(const char word[], const bool guessed[])
{
    printf("Word: ");
    for (int i = 0; word[i] != '\0'; i++) {
            if (guessed[word[i] - 'a']) {
                    printf("%c ", word[i]);
            }
            else {
                    printf("_ ");
            }
```

```c
        }
        printf("\n");
}
void drawHangman(int tries)
{
        const char* hangmanParts[]
                = { "      ___", " |              |",
                    " |              O", " |  /|\\",
                    " |      / \\", " |" };
        printf("\n");
        for (int i = 0; i <= tries; i++) {
                printf("%s\n", hangmanParts[i]);
        }
}
```