helper.c

```c
#include<string.h>
#include<stdio.h>
#include <stdlib.h>
#include"cs50.h"
#include"helper.h"

// each of our text files contains 1000 words
#define LISTSIZE 1000

// values for colors and score (EXACT == right letter, right place; CLOSE == right letter, wrong place; WRONG == wrong letter)
#define EXACT 2
#define CLOSE 1
#define WRONG 0

// ANSI color codes for boxed in letters
#define GREEN   "\e[38;2;255;255;255;1m\e[48;2;106;170;100;1m"
#define YELLOW  "\e[38;2;255;255;255;1m\e[48;2;201;180;88;1m"
#define RED     "\e[38;2;255;255;255;1m\e[48;2;220;20;60;1m"
#define RESET   "\e[0;39m"

// Promts the user to enter the guessed word
// Also error checks if the length of the word is of right size, otherwise re-promts the user to enter
string get_guess(int wordsize)
{
    string guess = "";

    // ensure users actually provide a guess that is the correct length
    // TODO #3

    do{
        guess = get_string("Input a 5-letter word: ");
    }while(strlen(guess) != 5);

    return guess;
}


// Compares the guessed word at each stage and adds a score value,
// Correct letter, rigth position = 2 (EXACT), Correct letter, wrong position = 1 (CLOSE), Wrong letter = 0 (WRONG)
int check_word(string guess, int wordsize, int status[], string choice)
{
    int score = 0;

    // compare guess to choice and score points as appropriate, storing points in status
    // TODO #5

    // iterate over each letter of the guess
    for(int i = 0; i < wordsize; i++){

        // iterate over each letter of the choice
        for(int j = 0; j < wordsize; j++){
            // compare the current guess letter to the current choice letter
            if(guess[i] == choice[j]){
                // if they're the same position in the word, score EXACT points (green) and break so you don't compare that letter further
                if(i == j){
                    score += EXACT;
                    status[i] = EXACT;
                    break;
                }else{
                    score += CLOSE;
                    status[i] = CLOSE;
                }
                // if it's in the word, but not the right spot, score CLOSE point (yellow)
            }
            // keep track of the total score by adding each individual letter's score from above
        }
    }
    return score;
}


// print_word - Prints the feedback to the user using different color codes (GREEN, YELLOW, RED)
// - Highlights the letters with GREEN for correct letter correct position,
// - With RED for wrong letters,
// - With YELLOW for correct letter wrong position.

void print_word(string guess, int wordsize, int status[])
{
    // print word character-for-character with correct color coding, then reset terminal font to normal
    // TODO #6

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

        if(status[i] == 2){
            printf(GREEN " %c " RESET, guess[i]);
        }else if (status[i] == 1){
            printf(YELLOW " %c " RESET, guess[i]);
        }else{
            printf(RED " %c " RESET, guess[i]);
        }
    }

    printf("\n");
    return;
}

```