```
/**
 * game.c
 *
 *
 *
-------------------------------------------------------------------
-----------
 *
 * 1. main()
 * Create an interactive program that allows a user to guess a
number between 1-10.
 * default max number (volatile) = 10;
 * default min number (const) = 1;
 *
 * 1b. generate a random number between min to max
 *
 *
-------------------------------------------------------------------
-----------
 * 2. menu(): display in console
 * Press 1 to play a game
        Press 2 to change the max number
        Press 3 to quit
 * 2b. get user input option
 * 2c. switch: calling corresponding sub-menu
 * 2d validate user input??
 *
 *
-------------------------------------------------------------------
-----------
 * [option 1] play a game
 * program should prompt the user to enter a number:
 *
 * 3a. the userGuess is digit (between min to max)
 * if userGuess == random
 *                tell them they won then the program,
 *        should go back to the main menu (step 2)
 * else if userGuess < random,
 *                tell them they were too low, continue (3a)
 * else
 *                tell them they were too high, continue (3a)
 *
 * 3b. userGuess is 'q'
 * If they enter q instead of a number when prompted,
 * the game should end(NOT the program) and return to the menu (step
2)
 *
 * 3c. invalid input, go back [option 1]
 *
 *
-------------------------------------------------------------------
-----------
 * [option 2] change the max number,
 * 4a. prompt user input: tell them the max value they can set the
```

```
number:
 * the max value they can set the number is the max of random
generated number
 *
 * 4b. validate user input:
 * Make sure they do not enter a negative number or go above the max
value.
 *
 * 4c. valid input: save variable int user_max_number to file in a
new branch later
 *
 * 4d. invalid input: go back to 4a
 *
 *
-------------------------------------------------------------------
------------
 * [option 3] quit
 * complete the following three ==>
 * 5a. thank the user for playing,
 * 5b. print out the result of all the games played.
 *     print out the number of guesses required to win.
 * NOTE: They could quit a game if this occurs indicate game was
lost.
 * 5c. end the game.
 *
 *
-------------------------------------------------------------------
------------
 * Finally create a new branch – call it save_user_max_number
 * Save the users request for the max number in -->
 * which the program can remember the max number from the previous
time
 * NOTE: This is called persists, meaning we can recall data after
process is stopped
 *
 * @author: Suhuan Pan
 */

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

#define CONST_MIN 1
#define MAX 10

#define OPTION1 "Press 1 to play a game"
#define OPTION2 "Press 2 to change the max number"
#define OPTION3 "Press 3 to quit"


/*********** phototype ***********/
void menu();
void initialize_random (int current_max);
void reset_random ();
```

```
void quit();
void game();
void play();
void allocate();


int number_of_guess = 0;

int computerGuess;
int current_max; // default 10
FILE *fp;
char ch;



/*********** step 1 ***********/
int main() {

        printf("Welcome to an interactive program that allows a
user to guess a number.\n");

        fp = fopen("text.txt", "w");

        // step 2
   menu();

        return 0;
}



/*********** step 4 ***********/
// 4a. tell user the max number they can set
// 4b. while loop,
// do not enter a negative number or go above the max value.
void reset_random () {

        printf("\n--- [option 2] change the max number ---\n");

        printf("The maximum number you can set is: %d.\n\n", MAX);

        int reset_max;

        while(1) {

                printf("Please enter the max number you want: ");
      scanf("%d", &reset_max);

                if ((reset_max) < CONST_MIN) {
                        printf("Invalid input, negative number is
not allowed.\n\n");
                }

                else if ((reset_max) > MAX) {
```

```
                        printf("Invalid input, value exceed the
max value.\n\n");
                }

                // 1 <= reset_random <=  MAX
                else {
                        printf("You reset the upper bound of
gussing to: %d\n", reset_max);

                        computerGuess = ((rand() % reset_max) +
CONST_MIN);

        // printf("***** DEBUG ONLY: computer's new guess  = %d\n",
computerGuess);

                        if (reset_max > current_max) {
                                current_max = reset_max;
                                printf("current max change to
%d\n", current_max);
                                break;
                        }
                        break;
                }
        }

        return menu();
}


/*********** step 3 ***********/
void game() {

        printf("\n--- [option 1] play a game ---\n");

   // ****************************
        printf("Enter a number to guess, 'q'to quit the game:\n");
        char input;
        scanf(" %c", &input);


        // 3b
        if (input == 'q') {
                printf("Going back to main menu...\n");
                return menu();
        }


        // 3a
        int a = input - 48;
        printf("You enter: %c and convert to ASCII is %d\n", input,
a);
        ch = input;

        putc(ch, fp);
```

```
        if (a == computerGuess) {

                printf("You won the game!\n");
                number_of_guess += 1;

                return menu(); // go back to step 2
        }

        else if (a < computerGuess) {

                printf("Too low...\n");
                number_of_guess += 1;

                return game();
        }

        else if (a > computerGuess){

                printf("Too high...\n");
                number_of_guess += 1;

                return game();
        }

        else {
                printf("something wrong...");
                fclose(fp);
                EXIT_FAILURE;
        }

        return;
}




/*********** step 2 ***********/
void menu() {

        char option;

                // 2a
                printf("\n--- main menu ---\n%s\n%s\n%s\n",
OPTION1, OPTION2, OPTION3);
                printf("Please enter option number:\n");

                scanf(" %c", &option); // 2b

                if ((option != '1') && (option != '2') && (option !
= '3')) {
                        printf("You enter invalid number.\n");
```

```
                        menu();
                }

                else if (option == '1') {

                        printf("You enter: %c\n", option);

                        if (number_of_guess == 0) {
                                computerGuess = (rand() % MAX +
CONST_MIN);
                                // printf("***** DEBUG ONLY:
computer's new guess  = %d\n", computerGuess);
                                game();
                        }

                        else {
                                game();
                        }
                }

                else if (option == '2') {
                        printf("You enter: %c\n", option);
                        reset_random(); // step 4,
                }

                else // (option == '3') {
                {
                        printf("You enter: %c\n", option);
                        quit(); // step 5
                }

        return; // return to main(); optional
}



/*
 ** 5a. thank the user for playing,
* 5b. print out the result of all the games played.
*     print out Number of guesses required to win.
* NOTE: They could quit a game if this occurs indicate game was
lost.
* 5c. end the game.
 */
void quit() {

        printf("\n--- [option 3] quit ---\n");

        printf("Thank you for playing.\n");

        printf("The correct guessing is: %d\n", computerGuess);

        printf("Game Ended.\n");
        fclose(fp);
```

```
        fp = fopen("text.txt", "r");
        printf("Display result ...");

        int index = 0;

        while ( (ch = getc(fp)) != EOF) {
                printf("\n%d: ", index);
                printf("%c", ch);
                index += 1;
        }
        printf("\nCurrent max is: %d\n", current_max);
        fclose(fp);


        printf("\nCurrent max is: %d\n", current_max);
        printf("\n\n");

        fp = fopen("user_max.txt", "w");

        ch = current_max + 48;

        putc(ch, fp);
        fclose(fp);

        return; // return to menu()
}
```