#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

#include <time.h>

#include <unistd.h>

typedef struct Player {

    char name[30];

    float luckyRatio;

    int guessedNumber;

} Player;

void unluck();

void tryAgain();

void right();

// Function to generate the random number form 1000-9999

int setRandomNumber()

{

    srand(time(NULL));

    int result = (rand() % 9000) + 1000;

    return result;

}

// Get and check name

void checkInputName(char \*name) {

    int stringLength, spaceCount;

    do {

        stringLength = strlen(name);

        spaceCount = 0;

        // Check the length and spaces in the name

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

            if (name[i] == ' ') {

                spaceCount++;

            }

        }

        if (stringLength > 30 || spaceCount > 0||stringLength==0) {

            printf("\033[1;31m");

            printf("Invalid Player name. It must be < 30 characters and >=1 character and should not contain spaces.\n");

            printf("\033[0m");

            printf("Enter your name again:");

            gets(name);

        } else {

            break;  // Exit the loop if the name is valid

        }

    } while (1);

}

//Check number

void checkInputNumber(int \*a)

{

    while (\*a>9999||\*a<1000)

    {

        printf("\033[1;31m");

        printf("Invalid number. It must be 1000-9999\n");

        printf("\033[0m");

        printf("Enter your number again:");

        scanf("%d",a);

    }

}

//Check option of player

void checkOption(char \*a)

{

    while((\*a!='y')&&(\*a!='n')&&(\*a!='d')&&(\*a!='D')&&(\*a!='Y')&&(\*a!='N'))

    {

        printf("\033[1;31m");

        printf("Invalid option. Enter your option again\n");

        printf("Press 'y' to continue.\nPress 'n' to exit.\nPress 'd' to see top 5 players\n");

        printf("\033[0m");

        printf("enter your option again:");

        getc(stdin);

        scanf(" %c",a);

    }

}

//Function to compare the rondam number with user's number and print the results in the required format

void printComparisonResult(int a, int b) {

    if(a!=b)

    {

        unluck();

        sleep(0.2);

        tryAgain();

        sleep(1);

    }

    printf("Here is your result: ");

    for (int i = 1000; i > 0; i /= 10) {

        int digitA = (a / i) % 10;

        int digitB = (b / i) % 10;

        if (digitA == digitB) {

            printf("%d", digitA);

        } else {

            printf("-");

        }

    }

}

//Function to load information of player into file

void savePlayerToFile(Player \*player) {

    FILE \*file = fopen("E:/cex/top2LuckyPlayer.txt", "a+");

    if (file == NULL) {

        printf("Error opening file.\n");

        return;

    }

    fprintf(file, "%s %d %.2f%%\n", player->name, player->guessedNumber, player->luckyRatio);

    fclose(file);

}

//Function to read information from file and printf out top 5 players

void findAndPrintTopPlayers() {

    FILE \*file = fopen("E:/cex/top2LuckyPlayer.txt", "r");

    if (file == NULL) {

        printf("Error opening file.\n");

        return;

    }

    Player players[100];

    int numPlayers = 0;

    while (fscanf(file, "%s %d %f%%", players[numPlayers].name, &players[numPlayers].guessedNumber, &players[numPlayers].luckyRatio) == 3)

    {

        numPlayers++;

    }

    fclose(file);

    // Sort players based on lucky ratio in descending order

    for (int i = 0; i < numPlayers - 1; i++) {

        for (int j = i + 1; j < numPlayers; j++) {

            if (players[i].luckyRatio < players[j].luckyRatio) {

                Player temp = players[i];

                players[i] = players[j];

                players[j] = temp;

            }

        }

    }

    // Print top 5 players

    printf("\e[1;33m");

    printf("////////////////////////////////////////////////////////////\n");

    printf("//                    TOP 5 PLAYERS                       //\n");

    printf("////////////////////////////////////////////////////////////\n");

    printf("%-35s %-10s %-15s\n", "Player", "Number", "Lucky Ratio");

    printf("------------------------------------------------------------\n");

    printf("\e[0m");

    for (int i = 0; i < 5 && i < numPlayers; i++)

    {

        printf("%-35s %-10d %.2f%%\n", players[i].name, players[i].guessedNumber, players[i].luckyRatio);

        sleep(1);

    }

}

// Function for starting game

void printWelcomeScreen() {

    char start;

    printf("\e[1;35m");

    printf("#       #     #  #####  #    # #     #      #     # #     # #     # ######  ####### ######  \n");

    printf("#       #     # #     # #   #   #   #       ##    # #     # ##   ## #     # #       #     # \n");

    printf("#       #     # #       #  #     # #        # #   # #     # # # # # #     # #       #     # \n");

    printf("#       #     # #       ###       #         #  #  # #     # #  #  # ######  #####   ######  \n");

    printf("#       #     # #       #  #      #         #   # # #     # #     # #     # #       #   #   \n");

    printf("#       #     # #     # #   #     #         #    ## #     # #     # #     # #       #    #  \n");

    printf("#######  #####   #####  #    #    #         #     #  #####  #     # ######  ####### #     # \n");

    printf("\e[0m");

    sleep(1);

    printf("Let start\n");

}

int main() {

    printWelcomeScreen();

    char userOption;    // variable stand for user's option

    Player newPlayer;

    do {

        int randomNumber = setRandomNumber();

        int incorrectEntriesCount = 0;

        printf("Enter your name: ");

        gets(newPlayer.name);

        checkInputName(newPlayer.name);

        // allow people enter number until have a right number

        do {

            printf("Enter your number:   ");

            scanf("%d",&newPlayer.guessedNumber);

            checkInputNumber(&newPlayer.guessedNumber);

            incorrectEntriesCount++;

            printComparisonResult(randomNumber, newPlayer.guessedNumber);

            printf("\n");

        } while (newPlayer.guessedNumber != randomNumber);

        sleep(1);

        right();

        newPlayer.luckyRatio = 100.00 / incorrectEntriesCount;

        printf("\e[1;31m%s\e[0m guessed the right number \e[1;31m%d\e[0m with a lucky ratio of \e[1;32m%.2f%%\e[0m", newPlayer.name, randomNumber, newPlayer.luckyRatio);

        savePlayerToFile(&newPlayer);

        printf("\nPress 'y' to continue.\nPress 'n' to exit.\nPress 'd' to see top 5 players\nWhat is your option: ");

        scanf(" %c", &userOption);

        checkOption(&userOption);

        if(userOption=='n'||userOption=='N')

        {

            return 0;

        }

        else if(userOption =='d'||userOption=='D')

        {

            findAndPrintTopPlayers();

        }

        else

        {

        }

        getc(stdin);

    } while (userOption == 'y'||userOption=='Y');

    return 0;

}

void unluck ()

{

    printf("\e[1;36m");

    printf("\n\n\n                   \*\*                 \*\*             \n");

    printf("                  /\*\*                /\*\*      \*\*   \*\*\n");

    printf(" \*\*   \*\* \*\*\*\*\*\*\*  /\*\* \*\*   \*\*  \*\*\*\*\* /\*\*  \*\* //\*\* \*\* \n");

    printf("/\*\*  /\*\*//\*\*///\*\* /\*\*/\*\*  /\*\* \*\*///\*\*/\*\* \*\*   //\*\*\*  \n");

    printf("/\*\*  /\*\* /\*\*  /\*\* /\*\*/\*\*  /\*\*/\*\*  // /\*\*\*\*     /\*\*   \n");

    printf("/\*\*  /\*\* /\*\*  /\*\* /\*\*/\*\*  /\*\*/\*\*   \*\*/\*\*/\*\*    \*\*    \n");

    printf("//\*\*\*\*\*\* \*\*\*  /\*\* \*\*\*//\*\*\*\*\*\*//\*\*\*\*\* /\*\*//\*\*  \*\*     \n");

    printf(" ////// ///   // ///  //////  /////  //  //  //      \n\n\n\n\n");

    printf("\e[0m");

}

void tryAgain()

{

    printf("\e[1;36m");

    printf("\a\a\n\n\n \*\*\*\*\*\*\*\*\*\*                                               \*\*            \*\*\n");

    printf("/////\*\*///          \*\*   \*\*              \*\*\*\*\*           //            /\*\*\n");

    printf("    /\*\*     \*\*\*\*\*\* //\*\* \*\*     \*\*\*\*\*\*   \*\*///\*\*  \*\*\*\*\*\*   \*\* \*\*\*\*\*\*\*   /\*\*\n");

    printf("    /\*\*    //\*\*//\*  //\*\*\*     //////\*\* /\*\*  /\*\* //////\*\* /\*\*//\*\*///\*\*  /\*\*\n");

    printf("    /\*\*     /\*\* /    /\*\*       \*\*\*\*\*\*\* //\*\*\*\*\*\*  \*\*\*\*\*\*\* /\*\* /\*\*  /\*\*  /\*\*\n");

    printf("    /\*\*     /\*\*      \*\*       \*\*////\*\*  /////\*\* \*\*////\*\* /\*\* /\*\*  /\*\*  // \n");

    printf("    /\*\*    /\*\*\*     \*\*       //\*\*\*\*\*\*\*\*  \*\*\*\*\* //\*\*\*\*\*\*\*\*/\*\* \*\*\*  /\*\*   \*\*\n");

    printf("    //     ///     //         ////////  /////   //////// // ///   //   // \n\n\n\n\n");

    printf("\e[0m");

}

void right()

{   printf("\n\n\n\n");

    printf("\e[1;33m");

    printf("########  ####  ######   ##     ## ######## \n");

    printf("##     ##  ##  ##    ##  ##     ##    ##    \n");

    printf("##     ##  ##  ##        ##     ##    ##    \n");

    printf("########   ##  ##   #### #########    ##    \n");

    sleep(1);

    printf("##   ##    ##  ##    ##  ##     ##    ##    \n");

    printf("##    ##   ##  ##    ##  ##     ##    ##    \n");

    printf("##     ## ####  ######   ##     ##    ##    \n");

    printf("\e[0m");

    printf("\n\n\n\n");

}

INSTRUCTION:

1. Enter the player's name.

If the number of characters exceeds 30 or less than 1 character or has space, an error will be notified, prompting the player to re-enter their name until they have the right name.

1. Enter the guessed number.

The guessed number must be in the range of 1000 to 9999. If the entered number falls outside this range, the program will notify an error and request re-entry.

If the guessed number is incorrect, the program allows the player to enter a new number until the correct one is found.

Once the player discovers the lucky number, the program will display the player's name, the lucky number, and the lucky ratio.

1. The program will then prompt the player to choose whether to continue:

Press 'y' to continue playing.

Press 'n' to stop.

Press 'd' print out the top 5 players with the highest lucky ratios.

If the player makes a choice that is not in the above table, the program will notify an error and let the user re-enter