lab11

```c
// EE231002 Lab11. Academic Competition
// 108061112, 林靖
// Date: Dec. 7, 2019

#include <stdio.h>                  // Standard input and output library

typedef enum {FALSE, TRUE} Bool;    // Boolean type

struct STU {                        // structure definition for each students
    char fName[15];                 //      first name
    char lName[15];                 //      last name
    double math, sci, lit;          //      test scores
    double min;                     //      minimum subject score
} ;
};
struct STU list[100];               // Subjects were tested among 100 students

Bool Find_Max_and_Print(const int *rank, double score[100]);
                                    // Find and print out the award winner with
                                    // max score.   Return FALSE if have
                                    //              printed out all the winners.

/*******************************************************************************
 *  Read all the data of the students into list[100], and simultaneously,     *
 *  for each student, find the min score among three subjects.                *
 *  After reading all the data, the prizes "Grand", "Math", "Sci" and "Lit"   *
 *  were processed and printed separately.                                    *
 ******************************************************************************/
int main(void)
{
    int i;                          // Index for looping
    double score[100];              // Scores of candidate winners
    int rank;                       // Ranking of the winner

    scanf("FirstName LastName Math Science Literature\n");
                                    // To skip the first line in the data file

    for (i = 0; i < 100; i++) {                         // For each student,
        scanf("%s %s %lf %lf %lf\n", list[i].fName,     // Read all the data
                                     list[i].lName,     // into list[100].
                                    &list[i].math,
                                    &list[i].sci,
                                    &list[i].lit);
        list[i].min = list[i].math;                     // Find the min score
        if (list[i].min > list[i].sci)                  // among three subjects
            list[i].min = list[i].sci;                  // and save it to
        if (list[i].min > list[i].lit)                  // list[i].min
            list[i].min = list[i].lit;
    }

    puts("Grand Prize:");
    for (i = 0; i < 100; i++) {                         // For each student
        if (80 <= list[i].min)                          // with all scores >= 80
            score[i] = list[i].math + list[i].sci + list[i].lit; // Copy score.
        else                                            // Otherwise,
            score[i] = 0;                               //     flag ineligible.
    }
    for (rank = 1; Find_Max_and_Print(&rank, score); rank++) ;
                                                        // Print all winners out
    puts("Math Prize:");
    for (i = 0; i < 100; i++) {                         // For each student
        if (60 <= list[i].min && list[i].min < 80)      // with all scores >= 60
            score[i] = list[i].math;                    //     Copy score.
        else                                            // Otherwise,
            score[i] = 0;                               //     flag ineligible.
    }
    for (rank = 1; rank <= 10 && Find_Max_and_Print(&rank, score); rank++) ;
                                                        // Print all winners out
    puts("Science Prize:");
    for (i = 0; i < 100; i++) {                         // For each student
        if (60 <= list[i].min && list[i].min < 80)      // with all scores >= 60
            score[i] = list[i].sci;                     //     Copy score.
        else                                            // Otherwise,
            score[i] = 0;                               //     flag ineligible.
    }
    for (rank = 1; rank <= 10 && Find_Max_and_Print(&rank, score); rank++) ;
                                                        // Print all winners out
    puts("Literature Prize:");
    for (i = 0; i < 100; i++) {                         // For each student
        if (60 <= list[i].min && list[i].min < 80)      // with all scores >= 60
            score[i] = list[i].lit;                     //     Copy score
        else                                            // Otherwise,
            score[i] = 0;                               //     flag ineligible.
    }
    for (rank = 1; rank <= 10 && Find_Max_and_Print(&rank, score); rank++) ;
                                                        // Print all winners out
    return 0;   // Normal program termination
}

/*******************************************************************************
 *  Find the max score in score[100] given, and print out the data of the     *
 *  student with the max score. Return FALSE if have printed out all the      *
 *  winners. Otherwise, return TRUE.                                          *
 ******************************************************************************/
Bool Find_Max_and_Print(const int *rank, double score[100])
{
    int i;                                  // Index for looping
    double score_max = 0;                   // The highest score found
    int index_max;                          // Index of the highest score

    for (i = 0; i < 100; i++) {             // For each score,
        if (score_max < score[i]) {         // find the highest score
            score_max = score[i];           // and save it to score_max.
            index_max = i;                  // Save index of the highest score
        }
    }

    if (score_max == 0)                     // If have printed out all the
        return FALSE;                       // winners, return FALSE.

    printf("  %d: %s %s %.1f\n", *rank,                 // Print out the data
                               list[index_max].fName, // of the student with
                               list[index_max].lName, // the heghest score.
                               score_max);
    score[index_max] = 0;                   // Flag that this score
                                            //          has been printed out.
    return TRUE;                            // There may be scores that have not
}                                           // yet been printed, so return TRUE.
```

[Format] can be improved.
[Coding] lab11.c spelling errors: heghest(1)
[Extra] array score is not needed.

Score: 93