lab10

```
$ gcc lab10.c
$ ./a.out < ../lab10.dat
Grand Prize:
  1: Ava BROWN 285.2
  2: John PRICE 284.9
  3: Isaac WASHINGTON 276.4
  4: Leah YOUNG 267.3
  5: Samuel BENNETT 264.5
  6: Alexis JACKSON 261.8
Math Prize:
  1: Gabriella HILL 99.3
  2: Elizabeth ANDERSON 98.9
  3: Abigail WILSON 98.3
  4: Benjamin RAMIREZ 97.4
  5: Isaiah BUTLER 97.4
  6: Daniel MORGAN 97.1
  7: Alexa PEREZ 96.8
  8: Alexander ROGERS 95.8
  9: Anna HERNANDEZ 95.4
  10: Christian BROOKS 95.3
Science Prize:
  1: James PETERSON 98.7
  2: Isaiah BUTLER 98.3
  3: Abigail WILSON 98.1
  4: Carter HAYES 96.6
  5: Dylan BARNES 96.3
  6: Noah MURPHY 95.5
  7: Avery LOPEZ 94.0
  8: Sydney EVANS 93.1
  9: Caleb HENDERSON 93.0
  10: Nicholas COLEMAN 91.7
Literature Prize:
  1: Elijah JAMES 99.7
  2: Jack SIMMONS 99.6
  3: Michael MORRIS 99.4
  4: Natalie MARTIN 98.9
  5: Nevaeh SCOTT 98.4
  6: Alyssa MARTINEZ 97.8
  7: James PETERSON 95.9
  8: Logan TORRES 95.7
```

```
  9: Audrey EDWARDS 95.5
  10: Alexa PEREZ 93.2

CPU time: 0.00584252 sec
score: 93
o. [Output] Program output is correct, good
o. [Format] Program format can be improved
o. [Coding] lab10.c spelling errors: intialize(1), posistion(1)
```

lab10.c

```c
// EE231002 Lab10. Academic Competition
// 109061158, 簡佳吟
// Date: 2020/12/7

#include <stdio.h>
#define N 100

struct STU {                // structure definition for each students
    char fName[15];         // first name
    char lName[15];         // last name
    double math, sci, lit;  // test scores
    double total;           // total score
};
struct STU list[100];
Comments?

int main(void)
{
    char ch;                // read the first line
    int i, k;               // index for loop
    int serial = 1;         // serial number for each prize
    int grand[N] = {0};     // record the position for Grand Prize winner
    int subject[N] = {0};   // record the position for subject prize candidate
    double max;             // store the maximum
    int top;                // record the posistion of maximum
    int end = 0;            // for stop the loop

    while ((ch = getchar()) != '\n') ;  // read the first line and discard it

    for (i = 0; i < N; i++) {           // scan every line
        scanf("%s %s %lf %lf %lf", list[i].fName, list[i].lName,
                        &list[i].math, &list[i].sci, &list[i].lit);

    }


// select who is eligible for Grand Prize or Subject Prize
    // select who is eligible for Grand Prize or Subject Prize
    for (i = 0; i < N; i++) {
        if (list[i].math >= 82 && list[i].sci >= 82 && list[i].lit >= 82) {
```

```
39              list[i].total = list[i].math + list[i].sci + list[i].lit;
40              grand[i]++;
41          }
42          else if (list[i].math >= 60 && list[i].sci >= 60
43                      && list[i].lit >= 60 && !grand[i]) {
44              subject[i]++;
45          }
46      }
47
48  // sort Grand Prize winner and prompt
        // sort Grand Prize winner and prompt
49      serial = 1;                                 // reset serial
50      printf("Grand Prize:\n");                   // prompt
51      for (i = 0; i < N; i++) {                   // find the max
52          if (grand[i]) {
53              max = list[0].total;                // initialize max
54              for (k = 0; k < N; k++) {
55                  if (list[k].total > max) {
56                      max = list[k].total;        // store max
57                      top = k;                    // record the position
58                  }
59              }
60
61              printf("  %d: %s %s %.1lf\n",serial++, list[top].fName,
                printf("  %d: %s %s %.1lf\n", serial++, list[top].fName,
62                        list[top].lName, list[top].total);      // prompt
63              list[top].total = 0;                              // discard it
64                                                                // and find the
65                                                                // next max
66          }
67      }
68
69  // sort Math Prize winner and prompt
        // sort Math Prize winner and prompt
70      serial = 1;                                 // reset serial
71      printf("Math Prize:\n");                    // prompt
72      for (i = 0; i < N && !end; i++) {           // find the max
73          max = list[0].math;                     // initialize max
74          for (k = 0; k < N && !end; k++) {
75              if (list[k].math > max && subject[k]) {
76                  max = list[k].math;                 // store max
```

```
77                  top = k;                                // record the position
78              }
79
80          }
81          printf("  %d: %s %s %.1lf\n",serial++, list[top].fName,
            printf("  %d: %s %s %.1lf\n", serial++, list[top].fName,
82                          list[top].lName, list[top].math);   // prompt
83          list[top].math = 0;                                 // discard it
84                                                              // and find the
85                                                              // next max
86          if (serial > 10) end = 1;                           // choose the first
87                                                              // ten people
88
89      }
90
91  // sort Science Prize winner and prompt
        // sort Science Prize winner and prompt
92      serial = 1;                                 // reset serial
93      end = 0;                                    // reset end
94      printf("Science Prize:\n");                 // prompt
95      for (i = 0; i < N && !end; i++) {           // find max
96          max = list[0].sci;                      // intialize max
97          for (k = 0; k < N && !end; k++) {
98              if (list[k].sci > max && subject[k]) {
99                  max = list[k].sci;              // store max
100                 top = k;                        // record the position
101             }
102
103         }
104         printf("  %d: %s %s %.1lf\n",serial++, list[top].fName,
            printf("  %d: %s %s %.1lf\n", serial++, list[top].fName,
105                         list[top].lName, list[top].sci);    // prompt
106         list[top].sci = 0;                                  // discard it
107                                                             // and find the
108                                                             // next max
109         if (serial > 10) end = 1;                           // choose the first
110                                                             // ten people
111     }
112
113 // sort Literature Prize winner and prompt
        // sort Literature Prize winner and prompt
```

```
114     serial = 1;                                 // reset serial
115     end = 0;                                    // reset end
116     printf("Literature Prize:\n");              // prompt
117     for (i = 0; i < N && !end; i++) {           //find max
        for (i = 0; i < N && !end; i++) {           // find max
118         max = list[0].lit;                      // initialize max
119         for (k = 0; k < N && !end; k++) {
120             if (list[k].lit > max && subject[k]) {
121                 max = list[k].lit;              // store max
122                 top = k;                        // record the position
123             }
124
125         }
126         printf("  %d: %s %s %.1lf\n",serial++, list[top].fName,
            printf("  %d: %s %s %.1lf\n", serial++, list[top].fName,
127                                 list[top].lName, list[top].lit);    // prompt
128         list[top].lit = 0;                                          // discard it
129                                                                     // and find the
130                                                                     // next max
131         if (serial > 10) end = 1;                                   // choose the fi
    rst
    This line has more than 80 characters
132                                                                     // ten people
133
134     }
135
136
137     return 0;                                   // done and return
138 }
139
140
```