

EE231002 Introduction to Programming

Lab10. Academic Competition

Due: Dec. 12, 2020

A senior high school is holding an academic competition. Three subjects were tested among one hundred students. The subjects are Mathematics, Science and Literature. The test results have been consolidated into a file, `lab10.dat`. Grand prize will be given to all students with all scores greater than or equal to 82. In addition 30 subject prizes, 10 for each subject, will be given out. Students, who are not receiving grand prize and having all scores greater than or equal to 60, are eligible for the subject prizes. Ten students with the highest subject scores will be awarded the subject prize. Thus, the grand prize winners only win one prize, while the subject prize winners can win more than one subject prize. Your assignment is to write a C program to process the result file and print out the award winners.

The resulting score file has been made self explanatory so the first line explains the contents of each following lines. Thus, there are totally 101 lines in the file, `lab10.dat`.

Since we have learned `struct` last week. This assignment is a good opportunity for you to try out coding using structures. The following structure definition is recommended.

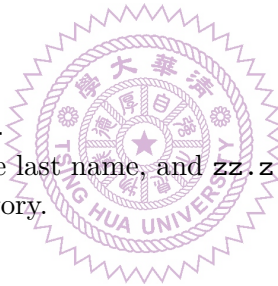
```
struct STU {                                // structure definition for each students
    char fName[15];                        // first name
    char lName[15];                        // last name
    double math, sci ,lit;                 // test scores
    double total;                          // total score
    double min;                            // minimum subject score
    int winTotal;                          // for winning Grand Prize
    int winSubj;                           // for winning a Subject Prize
};
struct STU list[100];
```

The first five members, from `fName` to `lit` are mandatory, while the remaining members are optional. You don't need to include them if you don't find them to be useful. One global structure variable `list`, which is a structure array of 100 elements, is also recommended for your coding.

The output format of your program should be as following:

```
$ ./a.out < lab10.dat
Grand Prize:
 1: xxx yyyyy zzz.z
 2: xxxx yyyyy zzz.z
 3: xxxxx yyyyyyyyy zzz.z
 4: xxxx yyyyy zzz.z
 5: xxxxxx yyyyyyy zzz.z
 ....
Math Prize:
 1: xxxxxxxxx yyyy zz.z
 2: xxxxxxxxx yyyyyyy zz.z
 3: xxxxxxxx yyyyyy zz.z
 ....
 ....
Science Prize:
 ....
 ....
Literature Prize:
 ....
 ....
```

where xxxxx is the first name, yyyy is the last name, and zz.z is the score, either total score or subject score depends on the award category.



Notes.

1. For the lab, **no sorting** needs to be performed. Since only few students will be selected, sorting on all students is not be an efficient approach.
2. Create a directory **lab10** and use it as the working directory.
3. Name your program source file **lab10.c**.
4. The first few lines of your program should be comments as the following.

```
// EE231002 Lab10. Academic Competition
// ID, Name
// Date:
```

5. After you finish verifying your program, you can submit your source code by

```
$ ~ee2310/bin/submit lab10 lab10.c
```

If you see a "submitted successfully" message, then you are done. In case you want to check which file and at what time you submitted your labs, you can type in the following

command:

```
$ ~ee2310/bin/subrec lab10
```

It will show the submission records of lab10.

6. You should try to write the program as efficient as possible. The format of your program should be compact and easy to understand. These are parts of the grading criteria.

