Course Work

1 Problem Description

You are asked to maintain a simple grade management system using the following data structure:

```
#define NAMELEN 10 // Students' names are no longer than 10 chars.
#define COURSES 4       // There are four courses in the system.

typedef struct stu_info
{       char stu_name[NAMELEN];
        int id, score[COURSES];
        float avgScore;
        struct stu_info *next;
} STU_INFO;
```

2 Initialization

The program offers a main menu to let the user choose different functions. Note that the program will return to main menu after each operation (except the exit option).

Example:

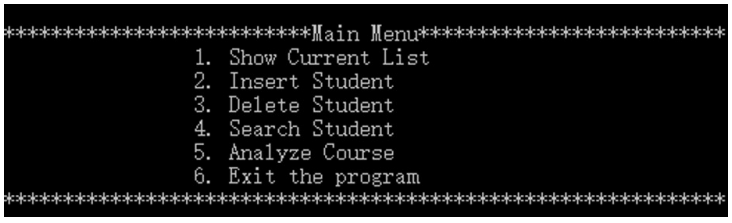

Figure 1Main Menu

There are 5 students in the default list. Your program should insert the following students and their scores to the list:

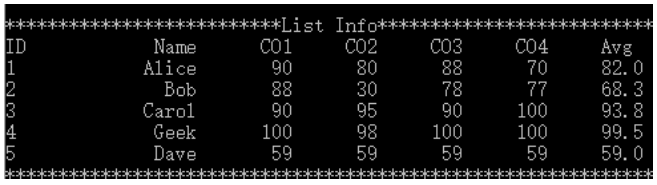

Figure 2 Default List

3 Main Functions

Your program MUST implement all the following functions:

1. The program offers a function to show the current list status.

   Example:

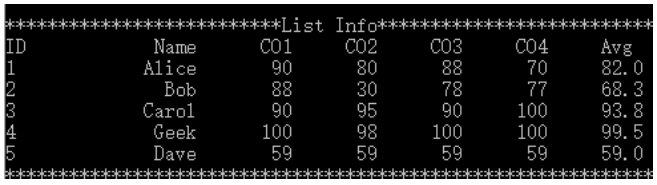

Figure 3 List Info Example

2. The program offers a function to insert a new student into the list. Users are asked to input the information of the new student.

   Example:

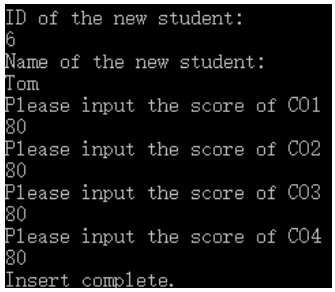

Figure 4 Insertion Example

   After insertion, the list has 6 students.

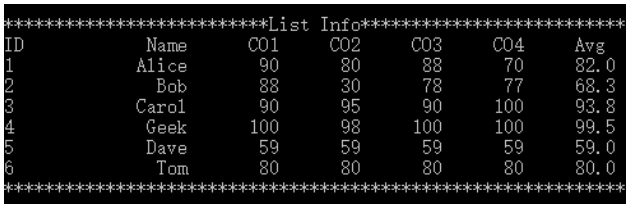

Figure 5 After Insertion

3. The program offers a function to delete a student given id.

   Example : Delete Alice (id = 1)

```
Please enter the id of the student to delete:
Id = 1
Delete operation complete
```

Figure 6 Delete Example

After deletion, the list has 5 students.

```
**************************List Info**************************
ID          Name      C01     C02     C03     C04     Avg
2            Bob       88      30      78      77     68.3
3          Carol       90      95      90     100     93.8
4           Geek      100      98     100     100     99.5
5           Dave       59      59      59      59     59.0
6            Tom       80      80      80      80     80.0
*************************************************************
```

Figure 7 After Deletion

4. The program offers a function to search a student by name. If the query student is found, the information of the students is shown.

   Example 1: query student found in the list:

```
Please enter the name of the student to search:
Name = Alice

Student Info:
ID          Name      C01     C02     C03     C04     Avg
1          Alice       90      80      88      70     82.0
```

Figure 8 Search for Alice

   Example 2: query student not found in the list:

```
Please enter the name of the student to search:
Name = adsf
Student <adsf> not found.
```

Figure 9 Search for asdf

5. The program offers a function to analyze a course given its course number.

   Example: (Take the default list as an example)

```
Please input the course id you want to analyze:
2

Course Info of C02 :
Max = 98.0
Min = 30.0
Avg = 72.4
PasssingRate = 0.60
```

Figure 10 Course Info Example

6. The program also provides a function to exit the program.

4 Optional Task

The program may also offer a mode named the sorted mode. Users are asked at the start of the program to enter an optional mode: default or sorted. If the user chooses the sorted mode, the initialled list is initialized sorted.

In the sorted mode, the list is kept the descending order in terms of students' average score.

(Hint) The students are sorted while insertion.

(Hint) There is **no need to sort the whole list** while each insertion.

(Hint) Here is a code frame:

```
STU_INFO *insert_node_sorted(STU_INFO *stu_list, STU_INFO *stu_node){

/*Insert a student information into the list in the descending order. Return a new stu_list */
/* Your Implementation Here */

return stu_list;
}
```

5 Report Requirements

Your programming report should include:

1. Title page with your name, student number, class number.
2. Flowchart to describe the algorithm of some important functions.
3. The screenshots of your program execution.
4. The C source code, attached in the appendix.

6 Standard of Grading

The coursework account for 10% of your overall grade for this course. The coursework will be marked 2 fold: program (8 points) and report (2 points).

Your programs will be marked based on (8 points):

- Correctness
- Readability
- Robustness
- Conciseness

7 Deadline

Submit your report before 12th Jan 2018