COMP2521 Final Exam

The University of New South Wales

Data Structures & Algorithms

COMP2521 Data Structures & Algorithms Final Exam

[Instructions] [Website] [C]
[Q1] [Q2] [Q3] [Q4] [Q5] [Q6] [Q7] [Q8]

Question 1 (15 marks)

Consider a small databse of student records in a text file like:

```
5012345:Smith, John:3707:65.0
9912345:Parameswaran, Sri:3707:99.0
5054321:Wang, David:3778:85.0
5012346:Smith, Janet:3778:75.0
6543210:Smith, James:3978:55.0
```

Each line has the following information about one student: zID, name, program code, WAM.

We have provided you with a program in `stu.c` that reads such text files via its standard input, stores the data into an array of `Student` records, and then displays the contents of this array.

If the above data was in a file called `data1` and if we were to compile and use `stu.c` unmodifed, it would produce the following output:

```
$ make
gcc -Wall -Werror -g   -c -o stu.o stu.c
gcc -o stu stu.o
$ ./stu < tests/data1
5012345 Smith, John                              3707 65.0
9912345 Parameswaran, Sri                        3707 99.0
5054321 Wang, David                              3778 85.0
5012346 Smith, Janet                             3778 75.0
6543210 Smith, James                             3978 55.0
```

The students records are in the same order as they were in the original file. We want to be able to sort the records two ways: by zID or by name.

Your task is to complete the `sortStudents` function which sorts the array of `Student` records in an order determined by one of its parameters. The function is defined as:

```
void sortStudents(Student *stu, int n, Ordering order)
```

The first parameter is a pointer to the first element in an array of `Student` records. The second parameter tells how many records are in the array. The third parameter determines which field is used for sorting and has two possible values: `BY_ZID` and `BY_NAME`.

The zID values are unique, so there is no issue in sorting them. However, names are not unique. If there are several students with the same name, they should appear as a group in the sorted array, ordered by zID.

You should *not* change any other part of `stu.c` except for the `sortStudents()` function, and any extra extra helper functions that you want to add. If you change the input and output functions or the `main()` function, then you will probably fail all of the tests.

The code for this question is in the `q1` directory, which contains:

- `stu.c` ... program for reading/sorting/displaying student records
- `Makefile` ...for building the program
- `tests/` ... directory containing test cases
- `run_tests.sh` ... script for running the tests

You ought to look at the data structures, the main program and the reading and writing functions, to familiarise yourself with the enviroment in which the `sortStudents()` function operates.

You can compile the program using the `make` command. This will give you an executable file called `stu`. Examples of running the `stu` command are given below.

Now,complete the `sortStudents()` function. You can define as many helper functions as you want. It requires around 15-30 lines of code to solve this problem, depending on how you solve it.

Submissions that don't compile are worth zero marks. Submissions that compile, but fail all tests are worth up to 7 marks. Submissions that compile and pass some tests are worth between 8 and 14 marks, depending on how close they are to a working solution. Submissions that compile and pass all tests are worth 15 marks.

Examples of how the program ought to behave when working correctly:

```
$ ./stu < tests/data2
3333333 Smith, John                              3645 75.0
5012345 Smith, John                              3707 65.0
5012346 Smith, John                              3778 75.0
6123456 Apple, Steve                             3648 85.5
6543210 Smith, John                              3978 55.0
$ ./stu name < tests/data2
6123456 Apple, Steve                             3648 85.5
3333333 Smith, John                              3645 75.0
5012345 Smith, John                              3707 65.0
5012346 Smith, John                              3778 75.0
6543210 Smith, John                              3978 55.0
```

You should also be able to devise your own test cases.

To help you check whether your program is working correctly, there is a script called `run_tests.sh` which will run the program against all of the tests and report the results. It will also add the output from your program into the `tests` directory; comparing your output against the expected output might help you to debug your code. You can run the testing script as:

```
$ sh run_tests.sh
```

Once your function is working (passes all tests), follow the submission instructions below. Even if it fails some (or even all) tests, you should submit because you can get *some* marks. If your program does not compile, or if you simply submit the supplied code, then your "answer" is worth zero marks.

Submission Instructions:

- Type your answer to this question into the file called `stu.c`
- Submit via: **`give cs2521 exam_q1 stu.c`**
  or via: Webcms3 > exams > Final Exam > Submit Q1 > Make Submission

End of Question