COMPSCI 1XC3 C01,C03

Assignment 4

(100 points)

Due date:April 8th, 2025

For assignment 4, we will refactor the code of assignment 3, and make the following improvements.

1. Define Struct based on Parallel Arrays

Please define `Course` struct, and `Student` struct in *course.h* and *student.h* . A course should have `id`, `name`, `capacity` and `students`. A student should have information including student `id`, `name` and `courses` taken.

2. Change Arrays to Lists

Please define `CourseListNode` struct, and `StudentListNode` struct in *course.h* and *student.h* . `Course` instances should be stored in the course list, and `Student` instances should be stored in the student list. Removed all the arrays used in assignment 3 as we are using lists in assignment 4. Change your main function accordingly, so that it still reads in *courseInfo.txt* and writes out *studentInfo.txt*.

3. Maintain Sorted Lists

Please make sure the `CourseList` is sorted based on course `id` alphabetically, and `StudentList` is sorted based on student `id` incrementally. Construct and maintain sorted lists when reading in the text file.

5. Adjust Student and Course Order in the Output File

When writing out the *studentInfo.txt*, make sure that student entries show up based on their `id` in ascending order. Under a student entry, all courses that the student takes should show up based on course `id` in alphabetic order.

6. Release Allocated Memory

After done using the lists with the output file written. Please remember to deallocate all the memories allocated on the heap.

7. Revise the main function

Revise the main function, so that it reads in file from argv[1], and writes out to file from argv[2].

```c
int main(int argc, char** argv)
{
  readCourseInfoFrom(argv[1]);
  writeStudentInfoTo(argv[2]);
  cleanup();
  return 0;
}
```

8. Clean up your code

Tidy up your code. You can leave relevant comments in the code, but do not leave unused code blocks from assignment 3.

9. Make sure your code compiles and runs

With your refactored *course.h* and *student.h* header files, please make sure your code compiles using `gcc main.c`. Please also make sure your code runs with `./a.out  courseInfo.txt studentInfo.txt` or `./a.exe  courseInfo.txt  studentInfo.txt`. TA will feed in new input files during grading.

10. Submit

Please submit *course.h*, *student.h* and *main.c* before or on April 8th, 2025. (Both header files need to be submitted, together with *main.c*)