CSI2131—File Management, Winter 2003, University of Ottawa

Assignment 2:
Cosequential Processing of Student Marks

Due: Tuesday March 11 at 10:30 a.m., Weight: 10%.

Introduction

The purpose of this assignment is that you get familiar with cosequential processing, and have a "hands on" experience with the concepts discussed in class. The main algorithmic approach to be used here resembles the ledger algorithm discussed in class (Section 8.2 of the textbook) but the application is about keeping student records and averages.

You are to write a program that processes information on students in SITE. We will assume on this assignment that at the end of the term the academic assistants for each undergraduate program (CEG, CSI, ELG, SEG) have a **transaction file** with the student marks for each course of this program. Note that the same student can be enrolled in many courses which can also be across the programs (say CSI2131 and SEG2101).

The transaction files have already been sorted by student number and for each student, by course. Each transaction record contains the following fields:

Field#	Field Name	Type	Length in bytes	Notes
1	Student Number	numeric	char[8]	can be converted to long int
2	Course Code	alphanumeric	char[7]	format LLLnnnn (CSI2131)
3	Credits (course)	numeric	char[1]	can be converted to int or float
4	Grade	alphabetic	char[2]	UofO style: F to A+

These transactions should be posted to a **master file** of student records. The master file is sorted by the field *Student Number* that uniquely identifies each student. The master file contains records with the following fields:

Field#	Field Name	Type	Length in bytes	Notes
1	Student Number	numeric	char[8]	unique key
2	Last Name	alphabetic	char[10]	
3	First Name	alphabetic	char[10]	
4	Credits	numeric	char[3]	number of credits completed
5	CGPA	numeric	char[4]	dd.d (eg. "10.0", " 9.8")

Specifically, the number of credits and CGPA of the student should be updated in the master file, as well as you should produce a report organized by student containing, for each student, all the courses talken by the student in this term with associated information: number of credits and mark for each course, total GPA (grade point average) for the term, and CGPA (cumulative grade point average) taking into account all terms. Note that the GPA and CGPA are averaged weighted by the number of credits.

You can read the master file *only once*. Consequently, you must process all transaction files *at the same time*. In addition, changes in a record of the master file must be recorded only once, after processing all transactions for certain student. Your implementation must be *independent of the number of transaction files* involved. We will also assume that the records on each transaction file are sorted by the field *Student Number*.

Note: all the fields containing numbers are recorded in text mode rather than binary mode.

Your program

We will provide a master file and 4 transaction files. They will be called: `master.txt`, `CEG.txt`, `CSI.txt`, `ELG.txt`, `SEG.txt`.

Your program should process the transaction files sequentially and in batch mode (no further input from the user while it executes). Call your executable program `marks.exe` and your updated master file `newmaster.txt`. We will use the following command to run your program:

```
marks master newmaster error.log CEG.txt CSI.txt ELG.txt SEG.txt
```

Note that the program must be general in that the number of transaction files provided may be any number. From the number of arguments in the command line, you can discover how many transaction files have been provided.
Do not read the entire transaction files or the master file all at once into main memory. In real situations, these files may be too large to fit in main memory.

There may be errors in the transaction files such as invalid student number, and your program should do some error checking. In this case, output the transaction and a short message to an error file.
The report should look like:

```
Student# 3456678, Smith, John
CSI2165 A+ 2 cr.
CSI2131 A- 3 cr.
SEG2101 B  3 cr.
GPA: 7.8  CGPA: 8.7

Student# 2345678, Einstein, Albert
CSI1101 D+ 3 cr.
CEG3151 B+ 4 cr.
GPA: 5.3  CGPA: 5.9
```

Assuming that John's previous CGPA was 10.0 with 5 credits, his new CGPA is calculated as shown above (average is weighted by credits and letter grades converted to number from 0 to 10).

How to hand in your assignment

Submit a zipped directory, as per the standards in the previous assignments, containing the following files:

- `marks.cpp`: the program that processes the transactions, updates the master file and generates the student report;
- `report.txt`: a report (textfile written on, say, notepad) containing the pseudo-code for your cosequential algorithm and any implementation details you find appropriate to explain (maximum 1 page).

Your program must meet the specifications given in this handout and in any further clarification given later on the web page.