Assignment #1: Formatted I/O (20 points)

This program reads student registrations to produce output suitable as a Register Request.
Command Line Arguments:

```
register -c studentRegistrationFile
```

Input:

```
Stream input file which contains many student records, each containing possibly many
registration requests.  There are three different kinds of lines of data for each student
record:
```

- Student Identification Information:
  - One line per request (separated by spaces)
  - szStudentId cGender szBirthDate szFullName
    7s 4s 11s 31s (may contain spaces)
  - Although szFullName is a maximum of 30 characters, it may contain spaces; therefore, you cannot simply use %30s. You will have to use a bracket format code using ^\n
- Student Record Information
  - One line per reservation request (separated by commas)
  - szMajor szEmail dGpa cInternationalStudent
    4s 31s 4d 1c
- Registration Request:
  - 0 to many registration requests per student (terminated by END in the course ID)
  - szCourseId
    12s

Files provided:

cs1713p1.h - include file
cs1713p1.c - program file which you must modify with your changes.
Multiple data files to check your error handling:
studentBad1.txt
studentBad2.txt
studentBad3.txt
studentRes.txt - this data file is used by your program for the output that you will upload into Blackboard.

Process:

1. Read a file of Data until EOF. For each student:
   - Read a data line containing the student's identification information.
   - Read a data line containing their record information.
   - Read possibly many (could be none) Registration Requests (until a course is read with END for the ID). There must be at least the END record.
   - Print the contents in a readable format. Examples:

```
****************** Student Registration Request  **********************
pte357 Pete Moss (M 1992/01/01)
CS pte357@my.uni.edu INTL:N GPA:3.21
   Course
   HIS1043.002
   GEO1013.005
   MAT1214.003
   CS1713.002
****************** Student Registration Request **********************
agt352 Pop Corn (M 1999/02/02)
MAT agt352@my.uni.edu INTL:N GPA:2.8
   Course
   MAT3013.001
   MAT3233.002
   HIS1053.004
```

Error Processing:

- Please include the errors you detect in stderr. For this program, the data input errors should cause your program to terminate.

Compiling:

- To compile, use:

```
gcc -o register cs1713p1.c
```

Grading:

- Your program must be written according to my programming standards.
- Your program should use the provided include file "cs1713p1.h".
- You must turn in your C code and its generated output which is based on the studentReg.txt data file.
- Modularity matters.
- Code which is mostly not working will receive less than 50% credit.
- If your output is manually manipulated in any manner, you will receive a zero.
- Even though your turned-in output should not have errors, error handling is required.
- Make certain your code works in Linux.

Note: Programming Assignment #2 will be a modification of this program.