[redacted]

Due date: Week 6 during your lab time.

Description
Your program will ask the user to type in the following information:
First Name:
Last Name:
ID#:
Email:

The program will terminate when a single . is entered as a First Name.

This information will be stored in the following structure type:

```
 struct student {
     int  recordCount;
    char *firstName;
    char *lastName;
    char *id;
    char *email;
}
```

recordCount is a number assigned at the beginning of the program starting at 1 and increasing by 1 for each student record entered. No name (first or last) will be more than 50 characters long but you must only store the exact number of characters needed. The id will be 9 characters in length (plus end of string). You must check to see if this is the case and that each character is a decimal digit (0 to 9). If it is not, the program will tell the user this and ask for the ID# again until it is entered correctly. There is no error checking on the email input.

Any number of students can be input to the program.

After all student records have been input, the program will print out all of the records sorted by ID# (smallest to largest). The output will look like the following:

091234567,Emma,Peel,mrspeel@avengers.co.uk
092345678,John,Steed,steed@avengers.co.uk

Sample Input and Output
$ ./enterStudents
First Name: Emma
Last Name: Peel
ID#: 091234567
Email: mrspeel@avengers.co.uk

First Name: John
Last Name: Steed
ID#: 092345678
Email: steed@avengers.co.uk
First Name:.
091234567,Emma,Peel,mrspeel@avengers.co.uk
092345678,John,Steed,steed@avengers.co.uk

Constraints

- No name can be greater than 50 characters in length
- No maximum number of student records

Program Structure

- The program is to be named *enterStudents.c*
- You must use a **makefile** to compile the code and to produce an executable called *enterStudents*.
- There will be no use of global variables.
- No debugging prints may appear during the execution of *enterStudents* when it is being graded.

Other Rules

- Your code must compile cleanly with no error or warning messages using –ansi and -Wall flags in gcc on the SOCS VM (Debian Linux)
- All work is to be done independently. If the TA deems that you have not done the coding independently then they will not grade the assignment.
- Code that cannot be compiled with **your** makefile will not be graded.

Make and Makefiles

Read the tutorial found here:
https://www.gnu.org/software/make/manual/html_node/Introduction.html#Introduction

Grading

When you are finished, ask the TA to grade your assignment. Be prepared to explain your code. The TA will test your code with random poems. There is no grading after the lab has ended – make sure that you finish in time for the TA to grade your assignment.