Requirements:

Your team is to design and implement a small class-roll maintenance system. For each student, the following data is needed:

* Name (up to 40 characters)
* USF ID (10 characters)
* Email (up to 40 characters)
* Grade of the presentation (numerical value from 0 (F) to 4 (A))
* Grade of essay (numerical value from 0 (F) to 4 (A))
* Grade of the term project (numerical value from 0 (F) to 4 (A))

The capabilities the system must support are:

* Read/write student data
* Add/delete students
* Retrieve student data based on a search by name, ID or email
* Update any or all data fields

The system consists of the main routine and various functions supporting the capabilities. There is no need (time) to implement a complicated user interface; a simple console-based text interface would be perfectly acceptable. The way the data is stored in a file must be made transparent to the user!

Input / Output Data:

StudentData.txt contains the input and output for the program and is a newline delimited file.

The same file is used in order to save added students during the program execution. The program writes to the file in a way so it can reread the data upon next execution.

Ex:

Robert Cunningham

U123456789

[rcunningham@mail.usf.edu](mailto:rcunningham@mail.usf.edu)

2

3

4

John Doe

U987654321

[jdoe@mail.usf.edu](mailto:jdoe@mail.usf.edu)

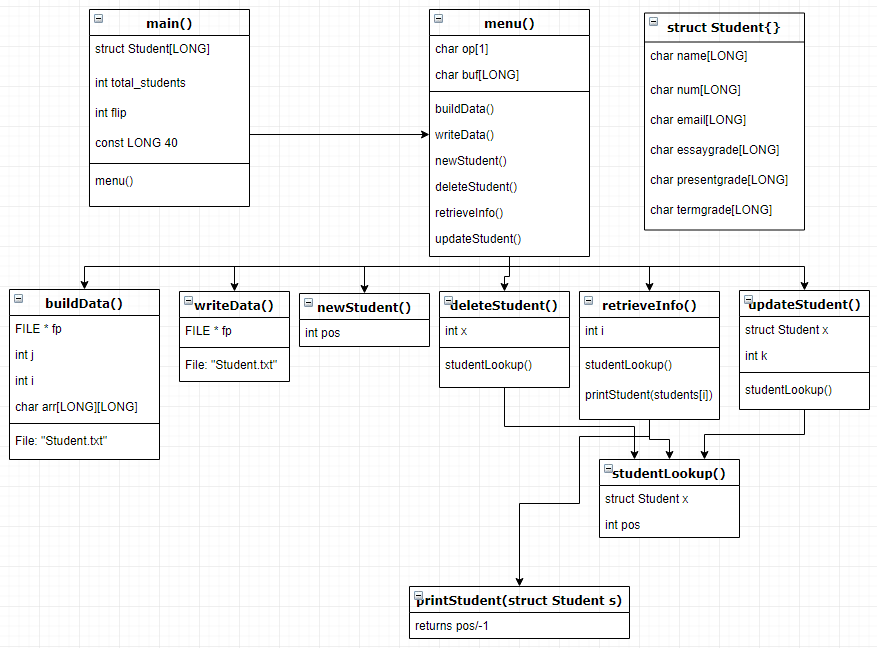
4

4

4

…

Flow Diagram of Functionality:



Bug Documentation

The code with the bugs embedded within it is called *main\_w\_bugs.c* and the corresponding text file with student information is called *StudentData\_bugs.txt*.

Bug #: 1

Bug Type: Improper Variable Initiation

Line(s): 19-20

**A picture containing object

Description automatically generated**

Details: *total\_students* is count variable we use to keep track of the number of students. Not initializing this variable to zero is what causes issues. Additionally, *flip* is a switch variable that essentially is being used as a Boolean value that flips between 0 and 1. This variable should be initialized at zero and it should be cast as a bool to stop possible problems that arise from assigning *flip* to a value other than 0 or 1.

Bug #: 2

Bug Type: Stray Characters

Line(s): 38

A black sign with white text

Description automatically generated

Details: Stray characters cause plenty of random bugs.

Bug #: 3

Bug Type: Flipped Comparator

Line(s): 78



Details: This will cause an out of range error. It should be *i < total\_students*.

Bug#: 4

Bug Type: Copy Error

Line(s): 118-120

A close up of a sign

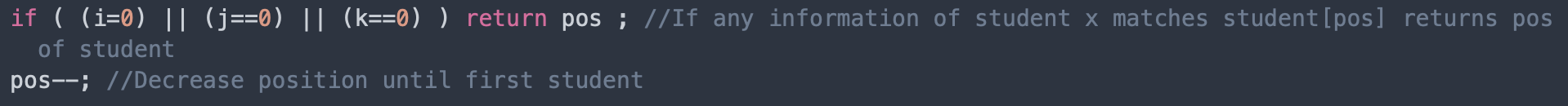
Description automatically generated

Details: This is a classic copy error. When lines are very similar we tend to copy and paste then change the parts that are different. The last two lines are supposed to add an email to the email field as the comment specifies. Instead it will change the name field.

Bug #: 5

Bug Type: Logical vs Assignment Operator

Line(s): 166



Details: Easy to make this mistake and mistype one equal sign instead of two. Most compilers will catch this and question you but the parenthesis around the argument *(i=0)* silences that warning.

Bug #: 6

Bug Type: String Compare

Line(s): 80

A close up of a sign

Description automatically generated

Details: These logical operations have not been overloaded anywhere in this code. In C one must use *strcmp* in order to compare strings properly.