

## EEL 5934/4930 Assignment 1

**Due:** Monday, January 23rd by 11:59 pm

You are going to implement three programs using the C programming language:

1. **Email Filter:** Input: A sequence of emails, Output: A sequence of calendar events  
The input will be read from the standard input and standard output will be used for output.  
Each line of the input will represent an email, which will be in the following format (we are abstracting away other fields such as from, to, etc.):

Subject: String

If the subject string is in one of the following format, then it is considered a calendar relevant event. The type of calendar events are as follows:

C,title,MM/DD/YYYY,HH:MM,location -> Create event with the given title, date, and time  
D,title,MM/DD/YYYY,HH:MM,location -> Delete event with the given title, date, and time  
X,title,MM/DD/YYYY,HH:MM,location -> Change the event with the matching title using the provided date, time, and, location

Please note that both title and location are required to be of length 10. If the actual string is shorter, it will be padded with space characters.

Note that Email Filter program will not take any of the actions specified above. Instead, it will write the subject string if it matches one of the event types.

As an example:

Sample Input (each email is on a separate line):

Subject: Hello

Subject: Greetings

Subject: C,Meeting ,01/12/2017,15:30,NEB202

Subject: Change in plans

Subject: X,Meeting ,01/12/2017,15:45,Larsen239

Subject: D,01/12/2017,15:45,Larsen239

Expected Output:

C,Meeting ,01/12/2017,15:30,NEB202

X,Meeting ,01/12/2017,15:45,Larsen239

Please note that the last email with the delete event is not well-formed as the title is missing. Therefore, it is not included in the output.

2. **Calendar Filter:** Input: A series of calendar events, Output: A sequence of pairs of dates and locations

We will assume that the calendar is empty (no events) when the program starts. It processes calendar events (each on a separate line) and updates the calendar. If an event changes the location of the earliest event for some date, then it outputs the date and the location in the following format:

MM/DD/YYYY:location

As in the case of Email Filter, location is exactly 10 characters (padded with whitespace if needed).

Both Email Filter and Calendar Filter should check for end of file (EOF) character while reading from the standard input. When testing your programs, if you enter the input on the terminal you can use CTRL-D to simulate EOF. Alternatively, you can save your input file and redirect it to your executable:

```
$ ./myprog < inputfile
```

3. **Location Updater:** Input: A sequence of emails, Output: A sequence of pairs of dates and locations

The input will be read from the standard input and the format is the same as that of the input for the Email Filter program. The output will be on the standard output and the format for the output is the same as that of the output for the Calendar Filter program. The program will process the incoming emails and update the calendars, which is assumed to be empty initially. If there is any change to the location of the earliest meeting on a date, then it will output the date along with the new location of the earliest meeting on that date. **Please note that such a change can occur due to a new event creation, time or location change for an existing event, or deletion of an event.**

Your solution should reuse Email Filter and Calendar Filter programs via the system calls fork, exec, pipe, and dup/dup2 system calls. Please note that using these two programs via copying & pasting into Location Updater will not receive any credit. Please see the submission instructions and the test case. Your solution will be tested on a Linux machine. So please make sure to test your solution on a Linux machine.