EEL 5733/4732 Assignment 1

Due: Monday, January 28th 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 the 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 types of calendar events are as follows:

C,title,MM/DD/YYYY,HH:MM,location -> Create an event with the given title, date, and time
D,title,MM/DD/YYYY,HH:MM,location -> Delete an 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

Both the title and the location are required to be of length 10. If the actual string is shorter, it will be padded with the space character. Note that the 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 and well-formed.

As an example:

Sample Input (each email is on a separate line):
Subject: Hello
Subject: Greetings
Subject: C,Meeting ,01/12/2019,15:30,NEB202
Subject: Change in plans
Subject: X,Meeting ,01/12/2019,15:45,Larsen239
Subject: D,01/12/2019,15:45,Larsen239

Expected Output:
C,Meeting ,01/12/2019,15:30,NEB202
X,Meeting ,01/12/2019,15:45,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 tuples of dates, times, and locations

The input will be read from the standard input and the standard output will be used for output. 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. When it processes a calendar event that sets/changes the location of the earliest event for that date, in addition to updating the calendar it also outputs the date, time, and the location of the earliest event. So note that if a calendar event does not change the location of the earliest event then nothing should be output as result of processing that calendar event. In the special case of a deleting all events of a date, time will be displayed as --:-- and location will be displayed as NA. Each line of the output will be in the following format:

MM/DD/YYYY,HH:MM,location

As an example:

Sample input 1:
C,Meeting ,01/12/2019,15:30,NEB202
X,Meeting ,01/12/2019,15:30,Larsen239

Expected Output:

01/12/2019,15:30,NEB202
01/12/2019,15:30,Larsen239

Sample input 2:
C,Class ,01/13/2019,10:30,NEB102
C,Meeting ,01/12/2019,15:30,NEB202
X,Meeting ,01/12/2019,15:30,Larsen239
C,Lab ,01/12/2019,11:30,Benton321

Expected Output:

01/13/2019,10:30,NEB102
01/12/2019,15:30,NEB202
01/12/2019,15:30,Larsen239
01/12/2019,11:30,Benton321

Sample input 3:

```
C,Class     ,01/13/2019,10:30,NEB102
C,Meeting   ,01/12/2019,15:30,NEB202
D,Meeting   ,01/12/2019,15:30,NEB202
C,Lab       ,01/12/2019,17:30,Benton321
```

Expected Output:

```
01/13/2019,10:30,NEB102
01/12/2019,15:30,NEB202
01/12/2019,--:--,NA
01/12/2019,17:30,Benton321
```

Sample input 4:

```
C,Class     ,01/13/2019,10:30,NEB102
C,Meeting   ,01/13/2019,15:30,Larsen239
```

Expected Output:

```
01/13/2019,10:30,NEB102
```

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

Both the Email Filter and the Calendar Filter programs should check for the 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 the EOF. Alternatively, you can save your input file and redirect it to your executable as follows:

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

3. **Location Updater**: Input: A sequence of emails, Output: A sequence of tuples of dates, times, 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. When it processes a calendar event that sets/changes the location of the earliest event for that date, in addition to updating the calendar it also outputs the date, time, and the location of the earliest event.

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