Lab6
Deadline: Nov 3

Requirements
You are required to write a program that reads the content of a file. The number of lines in the file is not known (it could be thousands of lines). The format of each line is:
FirstName LastName GPA Status TOEFL

Sample input:
Mary Jackson 4.0 I 60
Jack He 2.45 D
….
Mike Johnson 3.125 D
Jane Zhang 3.8 I 120

Implementation Details

1. Your program should read all the lines and store each line as a struct named DomesticStudent and InternationalStudent.
2. Your program should then write to an output file a list of students whose GPA is greater than 3.9 to a file. GPA can be up to 3 decimal places.
3. An international student has a TOEFL score ranging from 0-120. If the international student has a TOEFL score less than 70, then it should not be written to an output file. TOEFL is an integer.
4. DomesticStudent struct does not have the TOEFL field.
5. I stands for an international student and D stands for a domestic student in status column.
6. The order of students must be the same order as in the original file.
7. If the format of a student does not conform the specified format, your program must write an appropriate message to an output file and exit.
8. The first argument is the input file name and the second argument is the output file name. The run command will be ./a.out <input file> <output file> <option>
9. The option field is the command line is an integer (1, 2 or 3). Here are descriptions of options and outputs only those students that meet the requirements of the above.
    a. 1 only saves the filtered output of domestic students (no international students in the output file)
    b. 2 only saves the filtered output of international students (no domestic students in the output file)
    c. 3 only saves the output of all students
10. You must handle corner cases with an output messages that contains “Error: XX”. This must be written to an output file.
11. Assume that everyone has a first name and last name.
12. Every line must end with \n including the last line in the output with no extra trailing spaces.

Verify your output:

Option 1: Only Domestic Students with GPA > 3.9

If the input file is:

```
Mary Jackson 4.0 I 60

Jack He 2.45 D

Mike Johnson 3.125 D

Jane Zhang 3.8 I 120

John Den 4.1 D

Alice Smith 4.200 I 80
```

The output file should be:

```
John Den 4.100 D
```

Option 2: Only International Students with GPA > 3.9 and TOEFL >= 70

The output file should be:

```
Alice Smith 4.200 I 80
```

Option 3: All Students with GPA > 3.9 (Domestic and International with TOEFL >= 70)

The output file should be:

```
John Den 4.100 D

Alice Smith 4.200 I 80
```

Error Handling

If the input file contains a line that does not conform to the specified format, the output file should contain an error message and the program should exit. For example, if the input file is:

```
Mary Jackson 4.0 I 60

Jack He 2.45 D
```

Invalid line

Mike Johnson 3.125 D

Jane Zhang 3.8 I 120

John Den 4.1 D

Alice Smith 4.2 I 80

The output file should be:

Error: Invalid format.

How to Compile and Run
gcc lab6.c -o <output executable>
./<executable> <input file> <output file> <option>

Submission Files

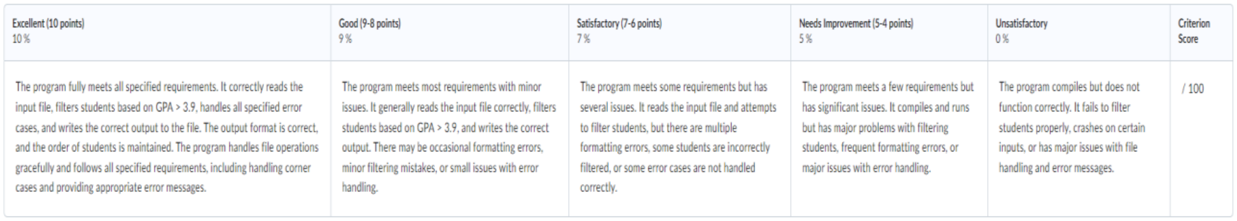
Instructions:

- Log in to the Learning Hub D2L.
- Navigate to the course's submission folders.
- Select the folder for "Labs Submission" then go to the specific lab, for example: Lab 6.
- Upload your program files (.c,.h, Makefile file) along with input.txt file created.
- Take a screenshot of your code and output and submit it as lab6.jpeg, or lab6.png
- Ensure that your submission is completed before the deadline.

Grading
Any grading failure due to not following instructions will result in 0.

- All files are submitted correctly using the instructions below.
- Generate a correct solution to the problem(s) in this lab. Three test inputs will be used.

Grading Rubric



Excellent (10 points) 10 %	Good (9-8 points) 9 %	Satisfactory (7-6 points) 7 %	Needs Improvement (5-4 points) 5 %	Unsatisfactory 0 %	Criterion Score
The program fully meets all specified requirements. It correctly reads the input file, filters students based on GPA > 3.9, handles all specified error cases, and writes the correct output to the file. The output format is correct, and the order of students is maintained. The program handles file operations gracefully and follows all specified requirements, including handling corner cases and providing appropriate error messages.	The program meets most requirements with minor issues. It generally reads the input file correctly, filters students based on GPA > 3.9, and writes the correct output. There may be occasional formatting errors, minor filtering mistakes, or small issues with error handling.	The program meets some requirements but has several issues. It reads the input file and attempts to filter students, but there are multiple formatting errors, some students are incorrectly filtered, or some error cases are not handled correctly.	The program meets a few requirements but has significant issues. It compiles and runs but has major problems with filtering students, frequent formatting errors, or major issues with error handling.	The program compiles but does not function correctly. It fails to filter students properly, crashes on certain inputs, or has major issues with file handling and error messages.	/ 100