

**Assignment 2****Deadline: 5:59PM on Nov 10**

In assignment 2, you will work in a group of maximum 3 of your choice.

**Requirements**

In this assignment, you are required to write a program that reads the content of a file and writes a sorted output to a file. The number of lines in the file is not known (it could be thousands of lines). The format of each line is:

```
>> cat input.txt
```

```
Mary Jackson Feb-2-1990 4.0 I 60
```

```
Jack He Feb-3-1990 2.45 D
```

```
Mike Johnson Sep-2-1980 3.125 D
```

```
Jane Zhang Mar-2-1970 3.8 I 120
```

You must use a merge sort to implement the sorting algorithm. Below is the sorting criteria where 1 is the first criteria to use and use the next criteria if the current criteria results in a tie.

1. Year of birth
2. Month of birth
3. Day of birth
4. Last name (alphabetical order)
5. First name (alphabetical order)
6. GPA
7. TOEFL (no TOEFL score is provided, then domestic students takes precedent)
8. Domestic > International

The birthday has the month written in a string. Only the following abbreviation will be used for input and must be used for the output.

Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

It must be an ascending order. For example using the sorting criteria 1, someone born earlier must be outputted first. All sorting criteria must follow this ascending order. Below are additional specifications of this assignment.

1. An international student has an integer TOEFL score ranging from 0-120.
2. DomesticStudent struct does not have the TOEFL field.
3. I stands for an international student and D stands for a domestic student in status column.
4. If the format of a student does not conform the specified format, your program must write the appropriate message to an output file and exit ("Error: XXX"). replace XXX with an appropriate message.

5. The first argument is the input file, the second argument is the output file name, and the third argument is the option (specification 6). The run command will be
6. `./<name of executable> <input file> <output file> <option>`
1. The option field in the command line is an integer (1, 2 or 3). Here are descriptions of options.
  - a. Option 1 only saves the sorted output of domestic students (no international students in the output file)
  - b. Option 2 only saves the sorted output of international students (no domestic students in the output file)
  - c. Option 3 only saves the sorted output of all students
2. You must handle corner cases including input format errors with the output message as in specification 4.
3. Assume that everyone has a first name and last name. No middle name. All names are case insensitive, but the output must have the same capitalization as the input.
4. GPA ranges from 0.0 to 4.3 and assume that the input only has up to 3 decimal places.
5. Birthday must be displayed with three letter abbreviation as shown above with the first letter being the capital.
6. Year of birth will range from 1950 to 2010

### Restrictions

- ☐ For any reason, if your code does not compile/run, it will result in 0.
- ☐ If you use any standard library functions other than `stdio.h`, `stdlib.h`, `string.h`, you must consult with me prior to using it.
- ☐ Every line must end with `\n` including the last line in the output with no extra trailing spaces.
- ☐ You are not allowed to have any additional information written in the output file except the output.

### Grading

For any submission that does not compile or run due to not following instructions will get 0 (no exceptions). This includes not writing A numbers.

- ☐ (1 point) Correct A number and following submissions instructions below
- ☐ (6 points) Generate correct solutions to the problem
- ☐ (4 points) Handle error cases to the problems

### Submission Files

- ☐ You must submit only one file named to Learning Hub: **a2.c** (case sensitive).
- ☐ Make sure to update your A numbers by inserting the following in the main function:
- ☐ `// A numbers of everyone. AXXXX_AXXXX_AXXX format.`  
`char *ANum = "A0000";`

```
FILE *outputFile = fopen(ANum, "w");
```

```
if (outputFile == NULL) {
```

```
        printf("Failed to create the output file.\n");  
        return 1;  
    }  
  
    fclose(outputFile);
```