Lab8
Deadline: In lab in the week of Apr 1

Requirements

Write a generic merge sort program in C that can handle sorting of short, int, float, char and string. Implement the sorting algorithm using a linked list data structure.

Implementation Details

1. The program should support sorting of short, int, float, char and string using a generic merge sort algorithm.
2. The input and output files names are specified through command line arguments e.g ./a.out input.txt output.txt
3. The input.txt file is provided to read the array:
   ```
   >input.txt
   1
   101,412,219,732,516,813
   E
   ```

   The first line indicates the data type 1: short, 2: int, 3: float, 4: char, 5: string and E specifies the end of file.
4. The sorted array must be written to an output text file e.g:
   ```
   >output.txt
   101,219,412,516,732,813
   ```

5. There will be corner case testing e.g if there is short type specified in the input file and it contains and int type data (beyond the short range) it should write “Error” without double quotes to the output file.

Restrictions

1. Do NOT add any printf function. When I run your submission, it should not output anything. If the program outputs anything when grading, it will automatically get 0. No regrades in this case no matter what. Even if you spend countless hours in this lab and everything works in your code, making anything print in this lab will result in 0. I will not accept any regrade request in this category.
2. Do NOT write anything except “Error” without double quotes.

How to Compile and Run

- The Makefile for lab is provided.
- The Makefile is supposed to work with lab8.c, input.txt, output.txt and ref.txt files so, make sure to name your files accordingly.
- Run the following command in vs code Terminal.

  ```
  make
  ```

  It should compile the code without any errors.

  ```
  make convert_input
  ```

It should convert the input.txt file to unix encoding.

```
make run
```

It should run the compiled code.

- Run the following command to delete the out file.
  ```
  make clean
  ```
- Run the following command to convert the generated output to unix encoding.
  ```
  make convert_output
  ```
  It should convert the output.txt file to unix encoding.
- Run the following command to check your output with provided ref file.
  ```
  make check
  ```
- You are not supposed to make any changes in the Makefile.
- Make sure to install dos2unix utility using the following command:
  ```
  sudo apt-get install dos2unix
  ```
  For Mac
  ```
  brew install dos2unix
  ```

Grading

Make sure to test against those to ensure that your program output format works. Any grading failure due to not following specifications will result in 0. For full marks this week, you must:

- (1 point) Correctly submitting the code with ANumber and Name.
- (1 point) Corner case handling. You should not write anything except “Error”.
- (3 point) Generate a correct solution (including correct memory allocation and deallocation) to the problem(s) in this lab.

Submission Files

- You must submit only one file named to Learning Hub: **lab8.c**
- Submit it to learning hub before the lab session.