Assignment 1

Deadline: Due 11:59PM on 9 Feb 2024

Write a C program that reads two strings from the input file and checks whether they are anagrams of each other or not. An anagram is a word or phrase formed by rearranging the letters of another e.g listen and silent.

In this assignment, you will work in a group of three members of your own choice.

Requirements

- Implement a C program that reads two strings from a text file specified through command line arguments.
- Both strings are separated by **\n** character in the input file.
- The program should handle both uppercase and lowercase letters as equivalent.
- Ignore spaces and punctuation marks when checking for anagrams.
- You are guaranteed to have no numbers in the input file.
- A valid anagram has exactly same number of characters, both in subject and anagram.
- Implement error handling to check if the strings read from the file are valid (non-empty) and write appropriate error messages to output file.
- Write the output into a text file specified through command line arguments, it must be exactly “1! anagram” or “0! not anagram” without quotations as described in below example (case sensitive).
- You will write main function as well as other helper functions.
- You can validate your sample inputs if they are valid anagrams or not from this link https://wordsmith.org/anagram/anagram-check.cgi
- Any error case must not crash the program. It must output “error” to the output file (case sensitive).

Example

Case 1: Given the contents of input.txt:

>>input.txt

listen

silent

>><executable> input.txt output.txt

>>output.txt

1! anagram

Case 2: Given the contents of input.txt:

>>input.txt

This is a text.

This text is cool.

>><executable> input.txt output.txt

>>output.txt

0! not anagram

Restrictions

- Before using any standard library functions other than stdio.h, stdlib.h, and string.h, you must consult with me prior to using it.
- You are not allowed to use **printf** function throughout your code. You can write any error messages to the output file.
- The code must compile and run without errors; otherwise, it will receive a score of 0.

How to Compile and Run

- The Makefile for assignment1 is provided.
- The Makefile is supposed to work with a1.c, input.txt, output.txt and ref.txt files so, make sure to save 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`
  It should delete the specified file.
  `make convert_output`
  It should convert the output.txt file to unix encoding.
- Run the following command to test your output with provided relevant reference output.
  `make check`
- You are not supposed to make any changes in the Makefile.
- Make sure to install dos2unix utility, if not already installed using the following command:
  `sudo apt-get install dos2unix`
  For Mac
  `brew install dos2unix`

Grading

Any grading failure due to not following instructions including the assignment requirements will result in 0.

- (1 point) Submitting files correctly and including correct A number at line 1 of your code as comment. e.g //A012345 A012346 A012347

- (3 point) Handling error cases correctly.
- (6 point) Generate a correct solution to the problem(s).

Submission

- You must push only one .c file named: a1.c (case sensitive).
- Make sure to add your A number and your partners too at the top of a1.c as comments. Write your A number including leading 0's.