COSC2430 HW1: Array and Pointer

1. Introduction

You need to create a C++ program that can find out a subsequence of common characters from a list of strings. A subsequence is the combination of all common characters maintain the sequence from all the strings provided.

2. Program and input specification

You will get a list of strings from an input file. Each line will be considered as a single string. You need to find out the length of the common subsequence of characters from those strings which is the longest one.

Assumptions:

- The input file is a small plain text file; no need to handle binary files.
- Each input file can have maximum 4 strings.
- A string can have maximum 1000 characters.
- An input file may contain empty lines between strings. In this case, you will ignore the line.
- Character is anything in the string: letters, numbers, symbols, spaces, etc.
- The output should be exactly matched with the format in part 3.

The main C++ program will become the executable to be tested by the TAs. The Result file should be written to another text file (output file), provided with the Command line.

Notice the input and output files are specified in the command line, not inside the C++ code. Notice also the quotes in the program call, to avoid Unix/Windows getting confused. See part 4.

3. Input and Output

The input and output files are regular text files, where each line is terminated with a '\n' character. Each line will be treated as a string.

The output file will contain the length of the subsequence.

Output format:

- "Len: " + *the length of the subsequence*. Note, in "Len: ", there is a space after the colon (:).

All records should be processed sequentially from beginning to end. Note that, input and output, all are case sensitive. Please, see examples to be clarified about the format.

4. Program Execution:

The general call to the executable is as follows:

```
Subsequence "input=input1.txt;output=output1.txt"
```

You can call the example with another command line type,

```
Subsequence input=input1.txt output=output1.txt
```

5. Examples

Example 1 of input and output,

Input11.txt

```
abcd321ABCD
abcD123ABCd
```

Command line:

```
Subsequence input=input11.txt output=output11.txt
```

Output11.txt

```
Len: 7              // maximum length subsequence: abc2ABC is of length 7
```

Example 2 of input and output,

Input12.txt

```
Data Structures Using C++
Data Structures Using Java
COSC 2430 is Data Structures course
Data and Structures
```

Command line:

```
Subsequence input=input12.txt output=output12.txt
```

Output12.txt

```
Len: 15
```

Example 3 of input and output,

Input13.txt

```
Data Structures is an interesting course.
                                                    // empty line will not be counted
We love Data Structures course.
```

Command line:

```
Subsequence input=input13.txt output=output13.txt
```

Output13.txt

```
Len: 23
```

6. Requirements

- The output file should contain ONLY the length of the common subsequence in the format specified above. Failure to adhere to the format will result in you not getting credit. Please ensure you follow the correct format.
- Timeout is set to 2 seconds.
- The provided test cases are designed to help you; however, they are not all inclusive. You should test your program with your own test cases to ensure your program is robust and can handle all cases. Keep in mind the 2 second timeout.

7. Hand over your homework

- Homework is an individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. If you copy/download source code from the Internet or a book, it is better for you to acknowledge it in your comments, instead of the TAs detecting it. Code that is detected to be copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc.) will result in “Failure” for the course and being reported to UH upper administration.

- Homework 1 needs to be turned in to our Linux server, follow the link here http://cs.uh.edu/~rizk/homework.html to get started with the server, argument manager, and other homework policies/info.

- Make sure to create a folder under your root directory, name it hw1 (name need to be lower case), only copy your code to this folder, no test case or other files needed. If you use ArgumentManager.h, don't forget to hand over it too.