LAB 5 – Arrays and Pointers

Problem Description

1. Specification

Write a C program that contains two functions, one to input two *non-empty* strings and the other to compare them. The comparison returns an integer that indicates the ***first*** position (array index) where the two strings differ.

2. Implementation

- The program to be submitted is named `lab5.c`. **Use the given template `lab5.c`** and fill in your code. **Submit only file `lab5.c`.**

- The first function to be implemented is `myStrInput()`. See file `lab5.c` for its specification. Use `getchar` and a loop to read a line of characters, and store the input characters into the array. The loop terminates when a new line character '`\n`' is entered. The new line character '`\n`' is NOT part of the line (i.e., discard the new line character '`\n`').

- The second function to be implemented is `myStrCmp()`. See file `lab5.c` for its specification. The function returns an integer that indicates the ***first*** position (array index) where the two strings differ. Consider the following two special cases:
  - Two strings are equal. In that case the return value is –1.
  - One string is a substring of the other (e.g., CSE2031 and CSE2031E3.0). In that case, the return value is the length of the shorter string (i.e., the index of the null character in the shorter string).

- In both functions, **do not use array indexing** such as `s[i]`. **Use only pointers and address arithmetic** to manipulate the array elements. If you use array indexing in your code, your program will not be marked and given zero point.

3. Sample Inputs/Outputs

See file `lab5.io` for sample inputs and outputs.

Notes

- Complete the header in file `lab5.c` with your student and contact information.

- Do not modify the function definitions in file `lab5.c`.

- Assume that all inputs are valid. No error checking is required on inputs.