|  |  |
| --- | --- |
| CETN 3010 | PROGRAMMING IN C |

ASSIGNMENT #3 – Working with STrings

**Due: Friday, Oct 23rd, 2015 @ 1:30PM (ie. Before class begins)  
Value: 15% of course mark**

# General Requirements

1. **Create functions that perform the following string manipulations**
2. **subString function**

Write a function called ***subString*** to extract a portion of a character string. The function should be called as follows:

subString(source, start, count, result)

* *source* is the character string from which you are extracting the substring.
* *start* is the index number into *source* indicating the first character of the substring.
* *count* is the number of characters to be extracted from the *source* string



* *result* is an array of characters that is to contain the extracted substring

For example, calling *subString(“character”, 4, 3, result)* extracts the substring *“act”* from the string “character” and places the result in *result.*

1. **findString function**

Write a function called ***findString*** to determine if one character string exists inside another string. The first argument to the function should be the character string that is to be searched and the second argument is the string you are interested in finding. If the function finds the specified string, have it return the location in the source string where the string was found. If the function does not find the string, have it return -1. It should be called as follows:

int index = findString(source, stringToFind)



For example, calling *index =* *findString(“a chatterbox”, “hat”)* searches the string “a chatterbox” for the string *“hat”.* Because *“hat”* does exist inside the character string, the function returns 3 to indicate the starting position inside the source string where *“hat”* was found.

1. **removeString function**

Write a function called ***removeString*** to remove a specified number of characters from a character string. The function should take three arguments:

* the source string
* the starting index number in the source string
* the number of characters to remove

removeString(source, startingIndex, numberOfCharactersToRemove)

For example, calling *removeString(“the wrong son”, 4, 6)* results in the in the string being reduced to *“the son”.*

1. **insertString function**

Write a function called ***insertString*** to insert one character string into another string. The argument to the function should consist of:

* the source string
* the string to be inserted
* the position in the source string where the string is to be inserted

insertString(source, stringToInsert, insertPosition)

For example, calling *insertString(“the wrong son”, “per”, 10)* results in the in the string being changed to *“the wrong person”.*

1. **Naming your Project**

You must name your Visual Studio with the following naming convention.

*[Your Name]*\_CETN3010\_Assignment3

1. **Code Requirements**

Your recreation of this application must include the following techniques discussed in class:

1. Your functions should be stored in a separate file called stringFunctions.c and have an accompanying stringFunctions.h header file.

# Submission

You will submit this particular assignment as a zipped solution via Moodle. Your submission will be time stamped by Moodle. It is suggested that you confirm with your instructor that the submission was received.

# Instructions

1. **Don’t forget that a code review is part of this project. You will need to show your code to the instructor in class on the due date while going through an evaluation of the application’s functionality. You will need to explain how the code works and complete the code review part of the rubric.**
2. ***Late submissions will be subject to the late penalties laid out in the course outline.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Developing** | **Competent** | **Professional** | **Mark Awarded** |
| **subString Function** | Developing (0-1)  Little or no elements of this requirement are met. | Competent (2-3)  Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)  All elements of this requirement are met. Output is correctly displayed on the console screen. |  |
| **findString Function** | Developing (0-1)  Little or no elements of this requirement are met. | Competent (2-3)  Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)  All elements of this requirement are met. Output is correctly displayed on the console screen. |  |
| **removeString Function** | Developing (0-1)  Little or no elements of this requirement are met. | Competent (2-3)  Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)  All elements of this requirement are met. Output is correctly displayed on the console screen. |  |
| **insertString** | Developing (0-1)  Little or no elements of this requirement are met. | Competent (2-3)  Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)  All elements of this requirement are met. Output is correctly displayed on the console screen. |  |
| **Code Commenting** | Developing (0 pts.)  Little or no elements of this requirement are met. | Competent (1 pts.)  Some elements of this requirement are met. | Professional (2 pts.)  All elements of this requirement are met. Code is adequately commented with the programmer’s intention. |  |
| **Formatting/**  **Readability** | Developing (0 pts.)  Little attention to correct spacing, indenting and adequate use of curly braces. Code very difficult to read | Competent (1-2 pts.)  Some attention to correct spacing, indenting and adequate use of curly braces but could be improved. Code slightly difficult to read | Professional (3 pts.)  Code is well spaced and attention is paid to the overall layout of the code. Spacing, indentation and correct use of curly braces are all implemented in a professional manner. |  |
| **Student Number:** | | | **Student Name:** |  | Total:  ( /25) |