CMPT 130 - FIC 2022-03 - Assignment 3

Due Date: Wednesday 30th November 2022 at 11:55 PM

Instructor: Dr. Yonas T. Weldeselassie (Ph.D.)

Read this document in its entirety and carefully before you start anything and understand it. If you have any questions, don't hesitate to email me.

In this assignment, we will work with cstrings. In order to store cstrings, we will use C++ static or dynamic arrays of characters. You are not allowed to use the C++ string data type in this assignment. Read the restriction section below.

Consider the C++ program and function headers provided in the starter code text file uploaded onto Moodle together with this assignment.

Your Task

You are required to copy the given main program together with the function headers exactly as they are with no change whatsoever and then implement the functions so that the program works correctly. Each function is described in detail in the comments given together with the function header. The output of a sample run of the given program is provided below for your reference. You can write additional helper functions that you can call from within your functions if you like; but you are not allowed to make any changes to the given main program or the function headers. If you write any additional helper function, then it must be included in your submission. In order to help you avoid any typo and modification of any function signature, copy and paste the code provided in the starter code text file.

Restrictions

You are required to use C++ static or dynamic arrays of characters to store cstrings. You are NOT allowed to use any C++ string data type variable for any purpose. If you use a C++ string data type variable for any purpose in your program, you will automatically get zero mark. Moreover, you are NOT allowed to add any include directive. Specifically you are not allowed to include string, cstdlib or math libraries. Also you are not allowed to use any built-in cstring functions such as strlen, strcpy and strcmp.

Submission Format

You are required to submit your program online through Moodle. You will find a submission button for **Assignment 3** on **Moodle on Week 11** and you are required to upload the source code (.cpp) file of your C++ program containing the implementations of all of the functions. No assignment is submitted through email or hard copy; you must upload your work onto Moodle before the due date. Make sure to upload a CPP file.

Submission Due Date and Time

The due date to upload your program online through Moodle is **Wednesday 30th November 2022 at 11:55 PM**. Moodle will not allow you to upload after this date and time.

Marking

Your program will be tested under Microsoft Visual C++ 2010 Express and you are advised to test your program on the lab computers before submitting your assignment. A non working program will automatically get zero. A program that works but doesn't give right output or gives partial right output will lose marks depending how severe its shortcoming is.

Sample Run of the program

```
This program is designed to help you test your functions.

Testing cstrlen function
------------------------
The length of s1="irregular" is 9
The length of "" is 0

Testing countChars function
---------------------------
ch='r' is found in s1="irregular" 3 times.

Testing findChar function
-------------------------
ch='r' is found in s1="irregular" in the index interval [2, 9) at index 2
ch='r' is found in s1="irregular" in the index interval [3, 9) at index 8
ch='r' is found in s1="irregular" in the index interval [3, 8) at index -1
ch='r' is found in s1="irregular" in the index interval [0, 9) at index 1

Testing getCopy function
------------------------
A copy of "irregular" is s2="irregular"
A copy of s2="irregular" is s3="irregular"
s2 is modified to s2="" but s3 is still s3="irregular"
A copy of s2="" is s3=""

Testing rotateString function
-----------------------------
s4="asmara" rotated 30 times to the left becomes "asmara"
s4="asmara" rotated 23 times to the left becomes "aasmar"
s4="aasmar" rotated 2 times to the right becomes "araasm"
s4="araasm" rotated 42 times to the left becomes "araasm"
s4="araasm" rotated 41 times to the left becomes "maraas"
s4="maraas" rotated 14 times to the right becomes "asmara"
s4="asmara" rotated 44 times to the left becomes "maraas"
s4="maraas" rotated 50 times to the left becomes "raasma"
s4="raasma" rotated 45 times to the right becomes "smaraa"
s4="smaraa" rotated 10 times to the left becomes "aasmar"

Testing empty function
----------------------
Emptying s2="irregular" gives s2=""

Testing append function
-----------------------
Appending ch='t' to s2="" gives s2="t"
Appending ch='n' to s2="t" gives s2="tn"
Appending ch='t' to s2="tn" gives s2="tnt"
Appending ch='r' to s2="tnt" gives s2="tntr"
Appending ch='g' to s2="tntr" gives s2="tntrg"
Appending ch='v' to s2="tntrg" gives s2="tntrgv"
Appending ch='b' to s2="tntrgv" gives s2="tntrgvb"
```

```
Appending ch='s' to s2="tntrgvb" gives s2="tntrgvbs"
Appending ch='i' to s2="tntrgvbs" gives s2="tntrgvbsi"
Appending ch='p' to s2="tntrgvbsi" gives s2="tntrgvbsip"
Appending ch='g' to s2="tntrgvbsip" gives s2="tntrgvbsipg"
Appending ch='n' to s2="tntrgvbsipg" gives s2="tntrgvbsipgn"
Appending ch='n' to s2="tntrgvbsipgn" gives s2="tntrgvbsipgnn"
Appending ch='k' to s2="tntrgvbsipgnn" gives s2="tntrgvbsipgnnk"
Appending ch='i' to s2="tntrgvbsipgnnk" gives s2="tntrgvbsipgnnki"
Appending ch='a' to s2="tntrgvbsipgnnki" gives s2="tntrgvbsipgnnkia"
Appending ch='r' to s2="tntrgvbsipgnnkia" gives s2="tntrgvbsipgnnkiar"
Appending ch='l' to s2="tntrgvbsipgnnkiar" gives s2="tntrgvbsipgnnkiarl"
Appending ch='t' to s2="tntrgvbsipgnnkiarl" gives s2="tntrgvbsipgnnkiarlt"
Appending ch='t' to s2="tntrgvbsipgnnkiarlt" gives s2="tntrgvbsipgnnkiarltt"

Testing append function
----------------------
Appending s2="tntrgvbsipgnnkiarltt" to s3="" gives s3="tntrgvbsipgnnkiarltt"

Testing removeChar function
--------------------------
Removing ch='p' from s2="tntrgvbsipgnnkiarltt" gives s2="tntrgvbsignnkiarltt"
Removing ch='h' from s2="tntrgvbsignnkiarltt" gives s2="tntrgvbsignnkiarltt"
Removing ch='f' from s2="tntrgvbsignnkiarltt" gives s2="tntrgvbsignnkiarltt"
Removing ch='t' from s2="tntrgvbsignnkiarltt" gives s2="ntrgvbsignnkiarltt"
Removing ch='l' from s2="ntrgvbsignnkiarltt" gives s2="ntrgvbsignnkiartt"
Removing ch='v' from s2="ntrgvbsignnkiartt" gives s2="ntrgbsignnkiartt"
Removing ch='b' from s2="ntrgbsignnkiartt" gives s2="ntrgsignnkiartt"
Removing ch='f' from s2="ntrgsignnkiartt" gives s2="ntrgsignnkiartt"
Removing ch='e' from s2="ntrgsignnkiartt" gives s2="ntrgsignnkiartt"
Removing ch='o' from s2="ntrgsignnkiartt" gives s2="ntrgsignnkiartt"
Removing ch='a' from s2="ntrgsignnkiartt" gives s2="ntrgsignnkirtt"
Removing ch='t' from s2="ntrgsignnkirtt" gives s2="nrgsignnkirtt"
Removing ch='q' from s2="nrgsignnkirtt" gives s2="nrgsignnkirtt"
Removing ch='d' from s2="nrgsignnkirtt" gives s2="nrgsignnkirtt"
Removing ch='i' from s2="nrgsignnkirtt" gives s2="nrgsgnnkirtt"
Removing ch='g' from s2="nrgsgnnkirtt" gives s2="nrsgnnkirtt"
Removing ch='d' from s2="nrsgnnkirtt" gives s2="nrsgnnkirtt"
Removing ch='w' from s2="nrsgnnkirtt" gives s2="nrsgnnkirtt"
Removing ch='q' from s2="nrsgnnkirtt" gives s2="nrsgnnkirtt"
Removing ch='i' from s2="nrsgnnkirtt" gives s2="nrsgnnkrtt"

Testing removeCharAll function
------------------------------
Removing all occurences of ch='l' from s3="tntrgvbsipgnnkiarltt" (length = 20) gives
s3="tntrgvbsipgnnkiartt" (length = 19)

Testing isEqual function
------------------------
s2="nrsgnnkrtt" and s3="tntrgvbsipgnnkiartt" are not equal
s2="nrsgnnkrtt" and s3="nrsgnnkrtt" are equal

Testing isAnagram function
--------------------------
```

```
s2="nrsgnnkrtt" and s3="nrsgnnkrtt" are anagrams
s2="nkrttnrsgn" and s3="nrsgnnkrtt" are anagrams
s2="nkrttnrsn" and s3="nrsgnnkrtt" are not anagrams

Testing zigzagMerge function
----------------------------
The zigzag merge of s2="vkrw" and s3="HOCKCBT" is s5="vHkOrCwKCBT"

Testing getSubString function
-----------------------------
A substring of s1="irregular" starting from index 1 with 6 characters is "rregul"
A substring of s1="irregular" starting from index 6 with 6 characters is "lar"
A substring of s1="irregular" starting from index 1 with 3 characters is "rre"
A substring of s1="irregular" starting from index 3 with 3 characters is "egu"
A substring of s1="irregular" starting from index 1 with 2 characters is "rr"

Testing isSubString function
----------------------------
s2="vkrw" is a substring of s2="vkrw"
"" is a substring of s2="vkrw"
s2="vkrw" is not a substring of ""
s2="vkrw" is not a substring of s3="wvkv"

Testing countWords function
---------------------------
There are 7 words in s5="A  a c  ztx  fb  h   Z"

Deleting heap memories
----------------------
Deleting s2. Done!
Deleting s3. Done!
Deleting s5. Done!

Press any key to continue . . .
```