ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

BLG102E - INTRODUCTION TO SCIENTIFIC AND ENGINEERING COMPUTATION
2019 -2020 Spring
SUPPLEMENTARY EXAM 1

Release Date:	07.09.2020
Duration:	**60 minutes**
Instructors:	Assoc. Prof. Cüneyd Tantuğ, Assist. Prof. Ayşe Tosun, Turgut Uyar

Download the source code and the test code given for this question, add your implementations to this source code, and **submit the modified source code back to Ninova. Do not modify the main function. Add your implementations into the functions for which the templates are given in the source file.**

Use proper indentation, meaningful identifier names, and mark the parameters as constant wherever possible.

In this question, you will work with "sized" strings which are NOT terminated with a '\0' character. Instead, they are represented using a structure (`sizedstr_t`) that contains the string content and the length of the string. Implement your functions to perform several string operations. Do not use arrays with fixed size to define the strings, the string must be exactly as long as needed and not more. Therefore you will need to allocate memory for the string content dynamically.

Complete the functions in the order given below. You are not allowed to add any header files, i.e. you are not allowed to use any string functions from the C standard library.

1. Implement the function `new_sizedstr` that takes a C string ('\0' terminated) and returns a sized string of the given struct type that has the same string content and the proper length field value.

2. Implement the function `sizedstr_len` that takes a sized string and returns its length.

3. Implement the function `print_sizedstr` that takes a sized string and prints it.

4. Implement the function `sizedstr_cat` that takes two sized strings and returns a third sized string by appending the second string to the first.

5. Implement the function `sizedstr_cmp` that takes two sized strings and compares them in lexicographic (dictionary) order. It must return -1 if the first string comes first, 1 if the second string comes first, or 0 if they are equal.

6. Implement the function `delete_sizedstr` that frees any dynamically allocated memory areas so that the program will not leak memory.

You are given a main function which invokes these string operation functions and prints their results depending on command-line arguments. Check the test file carefully to understand how the main program is called with multiple arguments. The first argument is a letter indicating which function is to be performed. The letter *p* indicates $\mathrm{print\_sizedstr}$, l indicates $\mathrm{sizedstr\_len}$, c indicates $\mathrm{sizedstr\_cat}$, and o indicates $\mathrm{sizedstr\_cmp}$. Note that in the concat case, the main function will print both the concatenated string and the length of the string. Depending on the function called, there might be one or multiple arguments after the letter. **Do not modify the main function.**