**Instructions:**

1. Create a MS VisualStudio solution named **Chapter10**.
2. Create projects according to the assigned problems. **For each projects, implement 2 versions (C-string and string object).**
3. Use techniques and experience that you have obtained from previous chapters and this chapter.
4. Validate ALL input.
5. Code and test your programs. They **MUST** be completed without syntax, logic and run-time errors.
6. Include comments of your name, date, and brief descriptions in all source codes. Comment precondition and postcondition on all of your user-defined functions.
7. Compress **Chapter10**folder into **ONE** zipped file.
8. Submit or re-submit your zipped file before its due date&time.

**STARTER CODE**

int main()  
{  
 do  
 {  
 system("cls");  
 cout << "\n\tCMPR121: Chapter 10 - Char, C-Strings and Strings - assignments by Prof Q";  
 cout << "\n\t" << string(80, char(205));  
 cout << "\n\t 3. Word Counter";  
 cout << "\n\t 6. Vowels and Consonants";  
 cout << "\n\t 8. Sum of digits in a string";  
 cout << "\n\t10. replaceSubString Function (20pts Extra Credits)";  
 cout << "\n\t12. Password verifier";  
 cout << "\n\t13. Date Printer";  
 cout << "\n\t" << string(80, char(196));  
 cout << "\n\t 0. Exit";  
 cout << "\n\t" << string(80, char(205));  
 cout << "\n";  
  
 switch(inputInteger("\tOption: ", 0, 20))  
 {  
 case 0: exit(0);  
 case 3: challenge3(); break;  
 case 6: challenge6(); break;  
 case 8: challenge8(); break;  
 case 10: challenge10(); break;  
 case 12: challenge12(); break;  
 case 13: challenge13(); break;  
 default: cout << "\n\tERROR: Invalid option.\n";  
 }  
 system("pause");  
 } while (true);  
  
 return 0;  
}

**3 - Word Counter**

Write a function that accepts a pointer to a C-string as an argument and returns the number of words contained in the string. For instance, if the string argument is “Four score and seven years ago” the function should return the number 6. Demonstrate the function in a program that asks the user to input a string then passes it to the function. The number of words in the string should be displayed on the screen. Optional Exercise: Write an overloaded version of this function that accepts a string class object as its argument.

**6 - Vowels and Consonants**

Write a function that accepts a pointer to a C-string as its argument. The function should count the number of vowels appearing in the string and return that number.

Write another function that accepts a pointer to a C-string as its argument. This function Should count the number of consonants appearing in the string and return that number.

Demonstrate these two functions in a program that performs the following steps:

The user is asked to enter a string.

The program displays the following menu:

Count the number of vowels in the string

Count the number of consonants in the string

Count both the vowels and consonants in the string

Enter another string

Exit the program

The program performs the operation selected by the user and repeats until the user selects E to exit the program.

**8-Sum of Digits in a String**

Write a program that asks the user to enter a series of single-digit numbers with nothing separating them. Read the input as a C-string or a string object. The program should display the sum of all the single-digit numbers in the string. For example, if the user enters 2514, the program should display 12, which is the sum of 2, 5, 1, and 4. The program should also display the highest and lowest digits in the string.

**10 - replaceSubstring Function**

Write a function named replaceSubstring. The function should accept three C-string or string object arguments. Let’s call them string1, string2, and string3. It should search string1 for all occurrences of string2. When it finds an occurrence of string2, it should replace it with string3. For example, suppose the three arguments have the following values:

string1: “the dog jumped over the fence”

string2: “the”

string3: “that”

**12-Password Verifier**

Imagine you are developing a software package that requires users to enter their own passwords. Your software requires that users’ passwords meet the following criteria:

The password should be at least six characters long.

The password should contain at least one uppercase and at least one lowercase letter.

The password should have at least one digit.

**13 - Date Printer**

Write a program that reads a string from the user containing a date in the form mm/dd/yyyy. It should print the date in the form March 12, 2018.