Object Oriented Programming – Assignment 3

Important Instructions:

1- Subscript operator and integer iterators **are ALLOWED**, **do not use offset notations and pointers iterators**.
2- Pass all the pointers by value unless you explicitly need a pointer to be changed in callee.
3- Make sure that you DO NOT consume any single extra byte.

Write a program that performs following string manipulation functions:

1. void StringConcatenate(c-string1, c-string2)

Write a function that takes two strings inputs and appends str2 at the end of str1. **Don not change the return type Void.** For example,

String 1: “Happy Birthday” (Input String 1 doesn’t have any extra space)
String 2: “ to you !”
After StringConcatenate,

String 1: “Happy Birthday to you !”
String 2: “ to you !”

2. Char StringTokens(char*)**

Write a function which takes a string and returns an array of words in the string. For example:

String: I am a student of OOP in FAST-NU
Function StringTokens returns:

```
I
am
a
student
of
OOP
In
FAST-NU
```

Hint: words are separated by spaces.
Note: Do not consume space of single extra character. Token printing is not part of this function.

3. Char* ReverseSentence(char*)

Write a function that takes a sentence and returns its inverse, **use previous functions to accomplish this task to get credit**. For example

String: “I am Pakistani”

After calling ReverseSentence

String: “Pakistani am I” (Return new string. Do not change the original string. Printing is not part of this function.)

4. int CompareString(char* cstring1, char* cstring2)

Write a function that takes two c-strings and returns following values:

	Return Value
Cstring1 < Cstring2	-1
Cstring1 = Cstring2	0
Cstring1 > Cstring2	1

Take any two strings, sort them alphabetically (ignore casing), this is how your function should compare the strings.

5. **Students List Functionality:** This includes following:
   a. Read Students’ names from data file, save in dynamically allocated array (Do not consume a single extra byte).
   b. Display List (before sorting) – void DisplayStringList(char** list)
   c. Sort the list using your CompareString function … BubbleSort(…)
   d. Display Sorted List

(Practice Problem - Submission not required.)
Void CompressString(char*)

Write a function that takes a string and if it finds more than one consecutive occurrences of a character in the string, it removes the extra occurrences. For example:

String: “a”

String after compression: “a”
String: “aaaaaaa”
String after compression: “a”
String: “bbabbbbbcccdddddddddddeffffg”
String after Compression: “babcdefg”
Note: Do not use any extra string inside the function.

Important Note:

- You cannot change the function prototypes given in the questions.
- You cannot use break or goto statements. Breaks are allowed in switch cases.
- Built-in string functions are not allowed. Use your own string helper functions wherever you need.
- Violation of any of instructions may result in ZERO credit or deduction of marks.
- Submit one running cpp file and your data file. Compressed files are not allowed in submission.

Create a main program and then test all of these functions. You must dynamically allocate and deallocate memory to all the strings in your program (except the temporary buffer). There should not be any memory leakages and dangling pointers in your program.

Sample Run:

```
Testing StringConcatenate:

String 1:           “I am a student”
String 2:           “ of OOP in FAST-NU”
After Concatenation:
String 1:                  “I am a student of CP in FAST-NU”
String 2:                   “of OOP in FAST-NU”
-----------------------------------------------------------------------------------------------------------

Testing StringTokens:

Tokens of String 1 are as follows:

I
am
a
student
of
CP
In
FAST-NU
-----------------------------------------------------------------------------------------------------------

Testing ReverseSentence

Reverse Sentence of String1 is:            “FAST-NU in OOP of student a am I”
-----------------------------------------------------------------------------------------------------------
Student List Functionality
Display Unsorted and Sorted List of Students given in Data.txt
```

Note: Read Input Strings from Data.txt (copy the data given below in your data file). Strings given in Data.txt are just samples. User can give any string in file (of 80 characters at max). Submit your data file along with the cpp file.

Data.txt

```
//String 1
I am a student

//String 2
 of OOP in FAST-NU

82 // TotalStudents
```

Yasoob Tahavi
Abdul moeez
Muhammmad Suhaib
Jarar Asif
Waleed Ikram
Suhaib Ahmad
Isbah Malik
Hassaan Mustafavi
Abdul Moiz
Moazzam Anwaar
Areeba Shahzadi
Maryam Saqib
Zaid Asif
Rida Amir
Ubaid Ur
Umer Habib
Abdur Rehman
Ghulam Mohyudin
Anzar Zahid
Haris Umer
Fajar Ejaz
Awab Mujtaba
Muhammad Ahmad
Hassan Ali
Hamza Amer
Ameer Humza
Abdul Tawab
Hina Fatima
Zafeer Tariq
Aatika Abid
Abdullah Zia
Haisem Naeem
Minahil Tariq
Muzammil Rasool
Maarib Ahmed
Talha Mohy
Muhammad Anas
Subayyal Saeed
Shahryar Ahmad
Abdul Arham
Obaid Ullah
Saad Parvez
Zakriya Tariq
Saad Chaudhry
Faran Ahmad
Mehdy Hasnain
Arham Shahzad
Raabia Baig

Umair Asim
Muhammad Zamin
Ahmad Aziz
Muhammad Mujtaba
Khadeeja Wasif
Saad Hussain
Rafia Karim
Farhan Bukhari
Hassan Jaffar
Asad Tariq
Halima Sadia
Abdullah Basim
Muhammad Tayyab
Muhammad Owais
Rabiya Irfan
Laiba Mubbashir
Muhammad Faizan
Ariba Arshad
Abdullah Suhail
Duaa Sohail
Muhammad Ahmad
Mahdiah Aqib
Hadi Ali
Hamza Ali
Rafeel Abdul
Mahad Hassan
Nabiha Tariq
Ahmad Sadeed
Mustaqim Afzal
Rafay Junaid
Hamza Mansoor
Abdul Rahim
Hassan Zubair
Abdul Hadi