**Object Oriented Porigramming (BSE-2A & BSE-2B) – Assignment 3**

**Spring 2021**

***Subscript operator and integer iterators are allowed, do not use offset notations and pointers iterators****.*

Write a program that performs following string manipulation functions:

1. **void StringConcatenate(char\* str1, char\* str2)**

Write a function that takes two strings inputs and appends str2 at the end of str1. 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 !”

1. **Char\*\* StringTokens(char\*)**

Write a function which takes a string and returns an array of words in it. 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.

1. **Char\*\* InverseStringTokens(char\*)**

Write a function which takes a string and returns its words in reverse order. Use previous function to accomplish this task. For example:

String: I am a student of OOP in FAST-NU.

Function returns Tokens in reverse order:

|  |
| --- |
| FAST-NU.  in  OOP  of  student  a  am  I |

Note: Do not consume space of single extra character.

1. **Char\* ReverseSentence(char\*)**

Write a function that takes a sentence and returns its inverse, use previous functions to accomplish this task. 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.)

1. **Void CompressString(char\*) (Submission not required.)**

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: “bbabbbbbcccddddddddddeffffg”

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.
* Do not use new/extra strings wherever mentioned in the questions.
* Violation of any of above 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. 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 CP in FAST-NU”  After Concatenation:  String 1: “I am a student of CP in FAST-NU”  String 2: “of CP in FAST-NU”  ------------------------------------------------------------------------------------------------------------  Testing StringTokens:  Tokens of String 1 are as follows:  I  am  a  student  of  CP  In  FAST-NU  ------------------------------------------------------------------------------------------------------------  Testing InverseStringTokens:  Tokens of the string in reverse order are as follows:  FAST-NU  in  CP  of  student  a  am  I  ------------------------------------------------------------------------------------------------------------  Testing ReverseSentence  Reverse Sentence of String1 is: “FAST-NU in CP of student a am I”  ------------------------------------------------------------------------------------------------------------  Testing CompressString  string 3: ”aaabbbcccdddeeefff”  String after Compression: “abcdef” |

**Note: Read Strings 1, 2 and 3 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 CP in FAST-NU  //String 3  aaabbbcccdddeeefff |