# Object Oriented Programming (BCS-2C, BCS-2D & BCS-2E) – Spring 2023
# Assignment 2

## Submission Instructions

**(Please read the following instructions carefully before submitting the assignment).**

1. Save your file with your Roll Number (format: xxL-xxxx_A2_partx.cpp).
2. There should be two cpp files submitted for this assignment. Do not submit multiple copies or empty files.
3. It should be clear that your assignment will **not get any credit** if:
    - The assignment is submitted **after the due date**.
    - Assignment is **plagiarized** from any source. (online/other students/etc.)
4. This assignment consists of **TWO SECTIONS,** first involves **MATRIX MANIPULATION,** and second involves **STRING MANIPULATION**.

---

## Part 1: (Matrix Manipulation)

Write a program that takes two matrices from user and performs following operations:

- Matrix Addition
- Transpose of a matrix
- Checks if a matrix is symmetric or not
- Interchange rows of a matrix

You are supposed to implement following functions:

1. **Int** InputMatrix( ifstream& fin, int& rows, int& cols)**
   **Description:** This function will take size of matrix from file, create a matrix dynamically, take matrix elements from file and return the matrix created. Subscript operator and Integer iterators are not allowed to **traverse** the matrix.
   **Note:** Data of Input File is given in the end of this file.
2. **Void OutputMatrix(int** matrix, cont int& ROWS, const int& COLS)**
   **Description:** Displays the matrix in proper format. Subscript operator and Integer iterators are not allowed to **traverse** the matrix.
3. **Int** AddMatrix(int** matrixA, int** matrixB, const int& ROWS, const int& COLS)**

**Description:** This function takes two matrices as parameters, adds them and saves the result in a newly created matrix R and returns the result. Subscript operator and Integer iterators are not allowed to **traverse** the matrix.

4. **int** TransposeMatrix(int** matrix, const int& ROWS, const int& COLS)**
   **Decription:** This function takes a matrix A, takes transpose of matrix A, saves the result in a newly created matrix and returns the result. Subscript operator is not allowed. Integer Iterators and offset notation ARE ALLOWED.
5. **Bool IsSymmetric(int** matrix, const int& ROWS, const int& COLS)**
   **Description:** This function takes a matrix as parameter with its size information and returns true if the matrix is symmetric and false otherwise. Call Transpose Matrix and compare both matricec, if matrix is equal to its transpose, it is symmetric. Subscript operator is not allowed. Integer Iterators and offset notation IS ALLOWED.
6. **Void InterchangeRows(int** matrix, const int& ROWS, const int& COLS )**
   **Description:** This function takes two row numbers and calls following function to actually interchange the rows.
7. **Void InterchangeRows( int*& row1, int*& row2 )//Swap**
   **Description:** This function interchanges two rows. You are NOT ALLOWED to iterate through rows and swap their values. Think of simple solution.

**Important Notes:**

- You cannot change the prototypes of the functions.
- You can use subscript operator to allocate and deallocate the memory.
- Your program should follow the exact sequence of Sample Run given below.
- Goto instruction is not allowed in your program.
- Submit only one running cpp file having all the functionality. DO NOT submit compressed files. Submit your data file as well to avoid any file related issues during evaluation.
- DO NOT take any input from user, we are taking input from file only.
- Violation of any of the above instructions may result in ZERO credit or marks deduction.

**Sample Run (with sample inputs):**

```
Matrix A =
1       2       3
4       5       6
7       8       9

Matrix B =
2       5       8
5       6       9
8       9       10
```

```
Matrix C =
2       3       4
5       6       7

A + B =
3       7       11
9       11      15
15      17      19

A+C =
Addition not possible.

Transpose of A =
1       4       7
2       5       8
3       6       9

Transpose of C =
2       5
3       6
4       7

Matrix A is NOT Symmetric

Matrix B is Symmetric

Interchanging Rows of Matrix A:
row1: 1         //Hard code this number
row2: 3         //Hard code this number

After Interchanging Rows Matrix A=
7       8       9
4       5       6
1       2       3
```

Note: These are only sample inputs. Your assignment may be evaluated on any value supported by data type.

InputFile.txt (Create a file InputFile.txt and paste following data in the file. Name of the file in your code should be “InputFile.txt”, it will be evaluated accordingly.) **Submit your data file in assignment submission with your only one running cpp file.**

```
//Format of data is given below
//Line1: Rows Cols
//Line2: <matrix[0][0]> <matrix[0][1]> <matrix[0][2]>...
//Line3: <matrix[1][0]> <matrix[1][1]> <matrix[1][2]>...
//Line4: Next Row and so on
```

```
//Matrix A
3 3
1 2 3
4 5 6
7 8 9

//Matrix B
3 3
2 5 8
5 6 9
8 9 10

//Matrix C
2 3
2 3 4
5 6 7
```

Input.txt

```
3 3
1 2 3
4 5 6
7 8 9
3 3
2 5 8
5 6 9
8 9 10
2 3
2 3 4
5 6 7
```

---

## Part 2: (String Manipulation)

**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** 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. Printing is not part of this function.

### 4. 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.)

### 5. 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.

### 6. 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

### Void CompressString(char*) (Practice Problem - 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.
- 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 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 OOP of student a am I”
-----------------------------------------------------------------------------------------------------
Student List Functionality
Display Sorted and Unsorted 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