Course Name:	Programming Fundamentals Lab	Course Code:	CL1002
Degree Program:	BCS	Semester:	Fall 2021
Exam Duration:	2 hours	Total Marks:	50
Paper Date:	22-Jan-2022	Weight	40
Section:	BCS-1A, BCS-1B, BCS-1C, BCS-1D, BCS-1E, BCS-1F, BCS-1G (9am – 12)	Page(s):	3
Exam Type:	Final		

Student : Name:_ _____ Roll No._ _____ Section:_ _____

Instruction/Notes:
- Attempt all questions.
- You must **NOT** use the built-in cstring functions like strlen etc. within your code
- Please make a different .cpp file for each question (e.g. question1.cpp for problem1 etc) copy all (.cpp) files in a folder named as your **rollnumber**. When your work is completed please paste this folder in the following location on xeon depending on your section.

  \\cactus\Xeon\Fall 2021\PF Lab Final Session1
- Please ensure that your work is properly submitted before leaving the lab. Empty/incorrect submissions will result in zero marks

Problem1 [Marks 30]

Question 1

Please note for doing this question you **cannot** use any of the builtin cstring library functions like strlen etc. and if you need a helper function you will have to write it yourself.

Write a program in C++ that accepts a sentence from the user, It passes this sentence to the following functions whose functionality is stated below:

1. **Trim**

   This function removes any leading or trailing spaces.

   so e.g. if the user enters " this is an exam ", the sentence after this function being called should be "this is an exam".

2. **RemoveExtraSpace**

   This function removes all the extra spaces present between the words. Your function may ignore any extra space present in the beginning or at the end of the sentence.

   So e.g. if the user had input:

   "This is an   exam."

   The output after removing extra spaces would be:

   "This is an exam".

3. DisplayCountofAlphabets

This function counts and displays the number of times an alphabet appears in the sentence case).

A sample main function and the output are given below

```cpp
int main()
{
        char sentence[101];
        cout<<"Please enter a sentence upto 100 characters long"<<endl;
        cin.getline(sentence, 100);

        Trim(sentence);
        cout<<"\nAfter Trim was called "<<endl<<sentence<<endl;

        RemoveExtraSpace(sentence);
        cout<<"\nAfter RemoveExtraSpace was called "<<endl<<sentence<<endl;

        cout<<"\nDisplaying count of alphabets "<<endl;
        DisplayCountofAlphabets(sentence);
        return 0;
}
```

```
C:\Windows\system32\cmd.exe

Please enter a sentence upto 100 characters long
    this        is      an      exam

After Trim was called
this        is      an      exam

After RemoveExtraSpace was called
this is an exam

Displaying count of alphabets
A appears 2 times
E appears 1 times
H appears 1 times
I appears 2 times
M appears 1 times
N appears 1 times
S appears 2 times
T appears 1 times
X appears 1 times
Press any key to continue . . .
```

This is page 2 of 3. The top-right corner shows a partial overlapping page with text about "Problem2 [Marks 20]", "Write a C++ function ca...", "square matrix is one...", "the two matrices...", "possible that...", "in the fo...", "The...".

Write a C++ function called **PatternMatch** which searches for a square matrix inside another square matrix. A square matrix is one in which the number of rows is the same as the number of columns. Your function should accept the two matrices and return true, if the second matrix is found inside the first one, and false otherwise. Note: it is possible that the second matrix occurs inside the first matrix after wrapping around horizontally or vertically, as shown in the following two pictures. The x's denote a possible region of the bigger matrix where a smaller one may occur. The x on the top-left corner of a smaller matrix is bold and underlined. In these examples the dimensions of the matrices are 6x6 and 3x3 respectively. Please note: these are only examples. Your function should deal with all possible cases for all possible dimensions, not just these ones.

Examples:

A sample main function that declares and initializes 2 square matrices of size 3 and 5 and then pass them as parameter to the function and display a message to the user specifying whether the smaller matrix was found within the bigger one, is given below

```cpp
#include<iostream>
using namespace std;
const int big=5;
const int small=3;

// some code here

int main()
{
        int data[big][big]={{3,2,7,3,3},{3,2,3,4,5},{1,1,3,4,5},{1,2,9,1,1},{2,3,7,2,2}};
        int pattern[small][small]={{1,1,1},{2,2,2},{3,3,3}};
        bool flag=PatternMatch(data,pattern);
        if(flag)
               cout<<"Pattern exists";
        else
               cout<<"Pattern does not exist";
        cout<<<endl;
        system("pause");
        return 0;
}
```

Best of Luck