National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Programming Fundamentals	Course Code:	CS
	Program:	CS	Semester:	Fall 2020
	Due Date:	08-10-2020	Total Marks:	100
	Open Date:	30-10-2020	Weight	4%
	Section:	H	Page(s):	5
	Exam Type:	Assignment # 2	Submission:	On Google Classroom

Objective: File Input/Output, Repetition and Loops.

Notes:

1. Understanding the question is part of assignment, so do not ask anyone else to understand the problem for yourself and tell you the solution. Do it yourself.
2. **The name of cpp files should be l20XXXX_q_no.cpp where XXXX is your roll number.**
3. Make a .zip file which will contain all cpp files and name that zip file as l20XXXX.zip and submit this zip file.
4. Always check the boundary values [Recall Problem solving].
5. Always use proper variable names and proper indentation for readability.
6. Always use comments to elaborate the code.
   you can comment in c++ through:
   // This is single line comment
   /* This is
   Multi line
   Comment */
7. Always follow the output and input format.
8. **CREATE ALL THE NECESSARY FILES FOR YOUR PROGRAMS BY YOURSELF.**
9. **DON'T involve in PLAGIARISM.**

Question # 1:

A pair of numbers is called amicable if their factors (excluding themselves) add up to each other. For example, the numbers **220** and **284** are amicable, because the factors of **220** are [1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110] and sum to 284, while the factors of **284** are [1, 2, 4, 71, 142] and sum to 220.

[ONLY CONSOLE INPUT & OUTPUT]

a) Write a C++ Program which takes any two positive integers from the user finds out whether the two numbers input by the user are amicable or not.

Sample I/O:

Enter first number of pair: 1184
Enter second number of pair: 1210
Output:
Yes, the pair is **amicable.**

[FILE INPUT & CONSOLE OUTPUT]

b) Write a C++ Program which reads a file which prints for all of the pairs that exist in files are amicable or not.

If the file **pairs.txt** contains:

220 284

1184 1200

2620 2924

5020 5564

6232 6368

The output would be:

Is the pair 220 284 amicable? Yes

Is the pair 1184 1200 amicable? No

Is the pair 2620 2924 amicable? Yes

Is the pair 5020 5564 amicable? Yes

Is the pair 6232 6368 amicable? Yes

Question # 2:

You want to assign grades for N students who take M courses (each student can take a different number of courses). For the example shown in output, each course has 2 quizzes, 2 assignments, 1 midterm and 1 final with the following weightages and maximum marks:

Quiz	5%	10 each
Assignment	10%	25 each
Midterm	35%	50
Final	50%	100

The assignment of grades based on the final score is as follows:

A	75 to 100
B	65 to 75
C	55 to 65
D	45 to 55
F	Below 45

a) Write a C++ program that asks for the value of N i.e. the number of students for which the data will be entered, and all of the marks obtained in the courses for those students. The output would be the total marks obtained and the grades assigned to those students in each course.

Sample Input/Output is given.

```
Enter number of students:       2
----- Student # 1 -----
Enter number of courses1:       2
        --- Course # 1 ---
        - Quiz # 1:     8
        - Quiz # 2:     7
        - Assignment # 1:       23
        - Assignment # 2:       19
        - Midterm:      38
        - Final:        85
        ---> Total:     93.4
        ---> Grade:     A
        --- Course # 2 ---
        - Quiz # 1:     5
        - Quiz # 2:     8
        - Assignment # 1:       14
        - Assignment # 2:       21
        - Midterm:      33
        - Final:        74
        ---> Total:     80.6
        ---> Grade:     A
----- Student # 2 -----
Enter number of courses2:       3
        --- Course # 1 ---
        - Quiz # 1:     4
        - Quiz # 2:     3
        - Assignment # 1:       15
        - Assignment # 2:       11
        - Midterm:      26
        - Final:        68
        ---> Total:     66.1
        ---> Grade:     B
        --- Course # 2 ---
        - Quiz # 1:     7
        - Quiz # 2:     2
        - Assignment # 1:       13
        - Assignment # 2:       9
        - Midterm:      22
        - Final:        65
        ---> Total:     61.2
        ---> Grade:     C
        --- Course # 3 ---
        - Quiz # 1:     8
        - Quiz # 2:     6
        - Assignment # 1:       17
        - Assignment # 2:       19
        - Midterm:      30
        - Final:        77
        ---> Total:     80.9
        ---> Grade:     A
Press any key to continue . . .
```

b) Extend the part-a in such a way:

1. Your program's inputs & prints all things on console same as above, but now along with above it stores all the inputs and outputs in a file as well.
2. Make a separate file for each student.
   Example for above output shown two files will be created, the content of files is given below.

Student_1.txt

```
Enter number of students:       2
----- Student # 1 -----
Enter number of courses1:       2
        --- Course # 1 ---
        - Quiz # 1:     8
        - Quiz # 2:     7
        - Assignment # 1:       23
        - Assignment # 2:       19
        - Midterm:      38
        - Final:        85
        ---> Total:     93.4
        ---> Grade:     A
        --- Course # 2 ---
        - Quiz # 1:     5
        - Quiz # 2:     8
        - Assignment # 1:       14
        - Assignment # 2:       21
        - Midterm:      33
        - Final:        74
        ---> Total:     80.6
        ---> Grade:     A
```

Student_2.txt

```
----- Student # 2 -----
Enter number of courses2:       3
        --- Course # 1 ---
        - Quiz # 1:     4
        - Quiz # 2:     3
        - Assignment # 1:       15
        - Assignment # 2:       11
        - Midterm:      26
        - Final:        68
        ---> Total:     66.1
        ---> Grade:     B
        --- Course # 2 ---
        - Quiz # 1:     7
        - Quiz # 2:     2
        - Assignment # 1:       13
        - Assignment # 2:       9
        - Midterm:      22
        - Final:        65
        ---> Total:     61.2
        ---> Grade:     C
        --- Course # 3 ---
        - Quiz # 1:     8
        - Quiz # 2:     6
        - Assignment # 1:       17
        - Assignment # 2:       19
        - Midterm:      30
        - Final:        77
        ---> Total:     80.9
        ---> Grade:     A
```

Question # 3:

Write a C++ program that reads two files **(A.txt, B.txt)** as input. It will then **Merge data of files A.txt and B.txt (which are sorted in descending order) in file C.txt**. Note that data in file C.txt should be copied in sorted descending order. You cannot use extra files and you cannot call or use any of sorting algorithms on file C.

Input/output:

A.txt:	B.txt:	C.txt:
4	9	9
3	5	5
2	4	4
1	3	4
		3
		3
		2
		1

Note: This program neither take input from user nor give output on console. You just run the program with two input files and a new file is created having the merged values.

Question # 4:

Write a C++ program called polynomial calculator which will perform some operations on polynomials. For each polynomial there are two important things which you need to store, first is degree of polynomial and second is coefficient of all terms. You can consider that maximum degree of all polynomials will be not more than 15. So, you will store degrees of polynomial in one file and coefficients in second file accordingly.

For example, a polynomial: $4x^6 -2x^3+ 6x^2+1$ would be stored as: Terms = 4 //Actual Size of both files

Coeff_File_1.txt:	Degree_File1.txt:
4	6
-2	3
6	2
1	1

Make sure that the degree file is sorted in descending order so that the highest power term appears first. Also, there should be no duplicates in the degree file and it.

Menu: Test the polynomial program via a menu-based system. Design a Menu which will take input an integer value from user and will execute all operations accordingly. Example of menu options for different values are given below:

0. **Exit the program**.
1. **FileInput & Print**: This will read polynomial # 1 from files (Coeff_File_1.txt, Degree_File1.txt) and will print that polynomial on console.
   Then output will be: **4x^6 -2x^3 +6x^2 +1**
2. **Equal:** This will read two polynomials **(Coeff_File_1.txt, Degree_File1.txt)** and **(Coeff_File_2.txt, Degree_File2.txt)** simultaneously and compare either the polynomials are equal or not.
3. **Add:** This will read two polynomials **(Coeff_File_1.txt, Degree_File1.txt) and (Coeff_File_2.txt, Degree_File2.txt)** simultaneously from files and add them and store in third file named as **(add_coeff_file.txt, add_degree_file.txt)**.
4. Subtract: This will read two polynomials simultaneously from files and add them and store in third file named as **(subtract_coeff_file.txt, subtract_degree_file.txt)**.

---------- **BEST OF LUCK** 😊 ----------