National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	**Programming Fundamentals**	**Course Code:**	**CL 118**
	Program:	**BS(CS)**	**Semester:**	**Fall 2018**
	Duration:	**2. hrs.**	**Total Points:**	**40**
	Paper Date:	**Thursday, 25th Oct 2018**	**Weight**	**25**
	Section:	**C**	**Page(s):**	**2**
	Exam Type:	**Lab Mid**		

Instruction/Notes: Taking some illegal online/offline help (i.e. cheating) might earn you an **F** grade in the entire course.

Question#1 [10 Marks]

Write a C++ program to input a number and check whether the number is Perfect number or not.

bool **isPerfect** (int number)

isPerfect returns true if the number is a perfect number and false otherwise. Use this function in the main to print appropriate message.

Perfect number is a positive integer which is equal to the sum of its proper positive divisors. A proper divisor of an Integer is any divisor without the number itself.

For example:

6 is the first perfect number
Proper divisors of 6 are 1, 2, 3
Sum of its proper divisors = 1 + 2 + 3 = 6.
Hence 6 is a perfect number.

Other examples of perfect number: 28, 496, 8128

Question#2 [15 Marks]

Write a C++ program that populates an array named myArray from file "**input.txt**" of size 15. The value of each element in myArray will be less than 10. Your task is to rotate the elements of myArray to right and output the result on console and file "**output.txt**".

You should also ask user how much rotation is required (i.e. rotate size) and rotate your array according to user's entered value.

void **rotateRight** (int arr[], int array_size, int rotateSize)

Sample Run

Input Array:	1	2	3	4	5	6	7	8
Rotate Size: **1**								
Output:	8	1	2	3	4	5	6	7

Input Array:	1	2	3	4	5	6	7	8
Rotate Size: **2**								
Output:	7	8	1	2	3	4	5	6

Hint: You can use another array for your help.

Question#3 [15 Marks]

Write a C++ program that can find the Complement of two initially populated arrays (Sets). Declare two int arrays A1 and A2 of size 10, now find the Difference/Complement between these two arrays (i.e. **A1-A2** = All Elements of set A1 that are not present in set A2) and store the result in a third array A3.

Note: The arrays A1 and A2 are **pre-sorted** and you have to use only **single loop** to solve this problem.

Sample Run:

A1: 1 3 6 9 10
A2: 2 3 10 12 14
Output:
A3: 1 6 9

A1: 2 7 8 15 87
A2: 15 63 71 87 91
Output:
A3: 2 7 8