National University of Computer and Emerging Sciences, Islamabad Campus

Course:	Data Structures	Course Code:	CS 2001
Program:	BS(Computer Science)	Semester:	Spring 2022
Due Date	21-April-2022 at 11:59 pm	Total Marks:	20
Type:	Assignment 2	Page(s):	2

Important Instructions:

1. Submit your source files in a zipped file named as your roll number, i.e., 20I-1111.zip.
2. You are not allowed to copy solutions from other students. We will check your code for plagiarism. If any sort of cheating is found, negative marks will be given to all students involved.
3. Late submission of your solution is not allowed.
4. You are not allowed to use any built-in library(string, math). You have to code from scratch.

Introduction:

In this assignment, given an expression of binary operators as input in the character array, you are required to design a solution in c++ for converting that expression in prefix, postfix and then evaluating that expression. Also you will check its validity (correct use of brackets and operators).

The intend is for the given input expression in character array

1. Check validity of the expression
2. Convert to Prefix expression
3. Convert to Postfix expression
4. Evaluate the given expression

You will consider the arithmetic operators (+ , - , / , *) and some additional operators given below.

Additional operators:

^ Operator: This operator is not available in c++ but your code must consider this operator as power operator. It should have precedence equal to multiplication (*) and division(/) operator. Its associativity is left to right.

% Operator: In c++ % operator is used as remainder operator and its precedence is equal to * and /. In your algorithm it should be remainder operator but its precedence should be greater than * and / operator.

Implementation

Declare a character array of size 100 and take an expression as input from the user. First, validate if the given expression is valid or not. E.g. (32-(45+2) is an invalid expression and should not be taken for further processing. However, if the expression is valid e.g. (120-(60+5)) then the program should first declare it to be a valid expression and then pass it on for change of notation. **Since in such of expressions, operands have variable length (like 45 has two characters and 120 has three), so you**

can display the postfix string separated by a space between two operands. The program should be able to produce such outputs for the given expression.

1. Input from the user=(120-(60+5))
2. Check validity and if valid
    a. Prefix: - 120 + 60 5
    b. Postfix: 120 60 5 + -
    c. Result: 55

3. if invalid Reenter expression or terminate