Assignment 03:

Implement a Parser for a Simple Context-Free-Grammar

Note: The last date of submission is 10-May-2023

Problem Statement:
In this assignment, you will be implementing a parser for a simple context-free grammar. The grammar describes a language of arithmetic expressions involving addition and multiplication, along with parentheses for grouping. The grammar is given below:

Grammar:
E → E + T | T
*E → T * F | F*
F → (E) | a

Sample Input:
*(a + a) * a*
Sample Output:
E
*T * F*
*F * F*
*(E) * F*
*(E + T) * F*
*(T + T) * F*
*(F + T) * F*
*(a + F) * F*
*(a + a) * F*
*(a + a) * a*

Sample Input:
*a * a **
Sample Output:
*Incorrect Structur*e

Requirements:
Your program may be written in automated parser generator like Flex-bison, Irony or you can write your own hand-written parser on Python, R, C, C++, Java, or C#. (If you would like to use a different programming language, please discuss it with me.)

Submission Instruction:

- *This assignment must be attempted in a **group with 2 to 3 members.***
- *Submit one hard copy at my office and soft copy through GCR.*
- *Marks will be awarded on the basis of individual **viva**.*

Penalties:
Work submitted any later than this will receive a mark of zero.

Plagiarism:
Very **severe penalties** will apply if you copy or otherwise reuse the work of others.