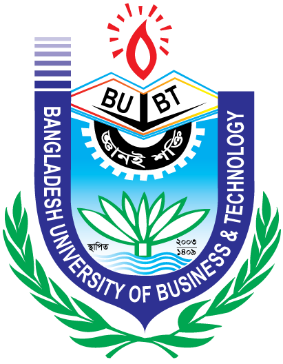
**BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY**

**(BUBT)**

****

**Lab Report**

Course Code : CSE 324

Course Title : Compiler Design Lab

Date of Submission :

Submitted By

Name : Aktaruzzaman

ID : 21222203031

Intake : 41

Section : 1

Submitted To

**Ms. Adeeba Anis**

Lecturer

Department of Computer Science & Engineering

Bangladesh University of Business and Technology (BUBT)

**Experiment No: 0**8

**Experiment No:** 9

**Experiment Name:** Recursive Descent Parsing Implementation

**Problem Structure:**

The objective of this experiment is to implement a recursive descent parser for a specific grammar. Recursive descent parsing is a top-down parsing technique where each non-terminal in the grammar is associated with a parsing function. The parser recursively calls these functions to parse the input string according to the grammar rules. The grammar is defined as follows:

E -> TE'

E' -> +TE' | ε

T -> FT'

T' -> \*FT' | ε

F -> (E) | a

**Procedure**

1. Define parsing functions for each non-terminal symbol in the grammar.

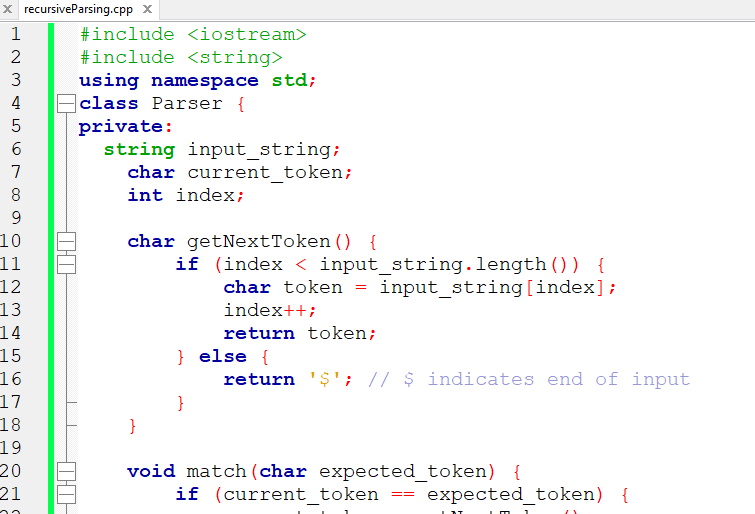
2. Implement parsing logic within each function based on the grammar rules.

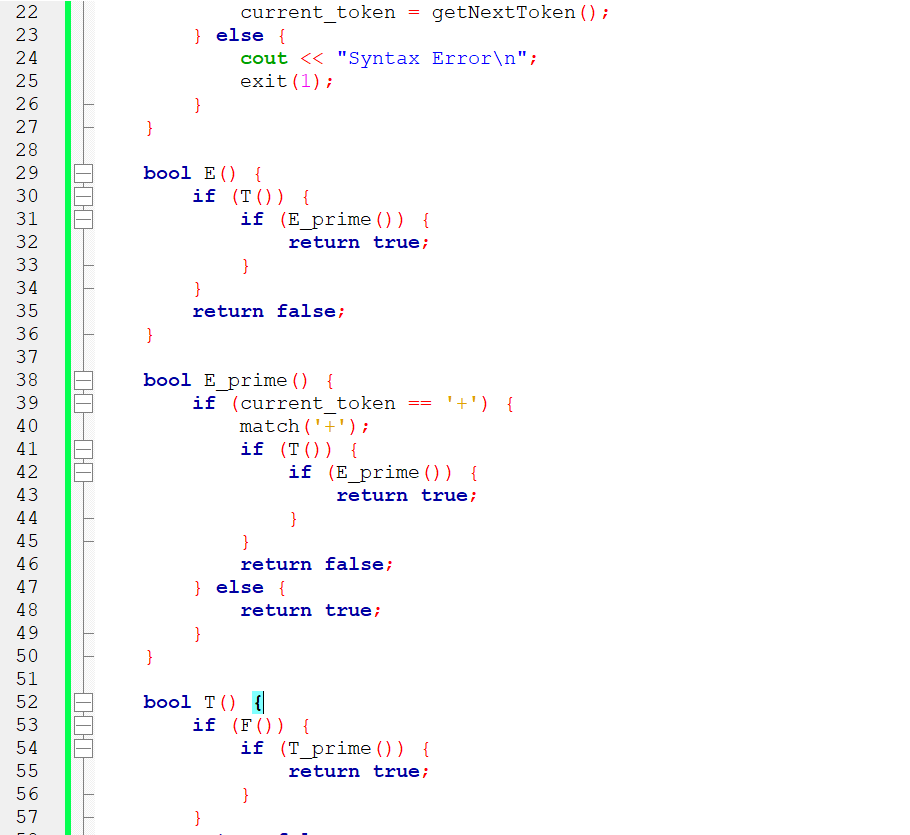
3. Use a `match` function to compare the current token with the expected token and advance to the next token if they match.

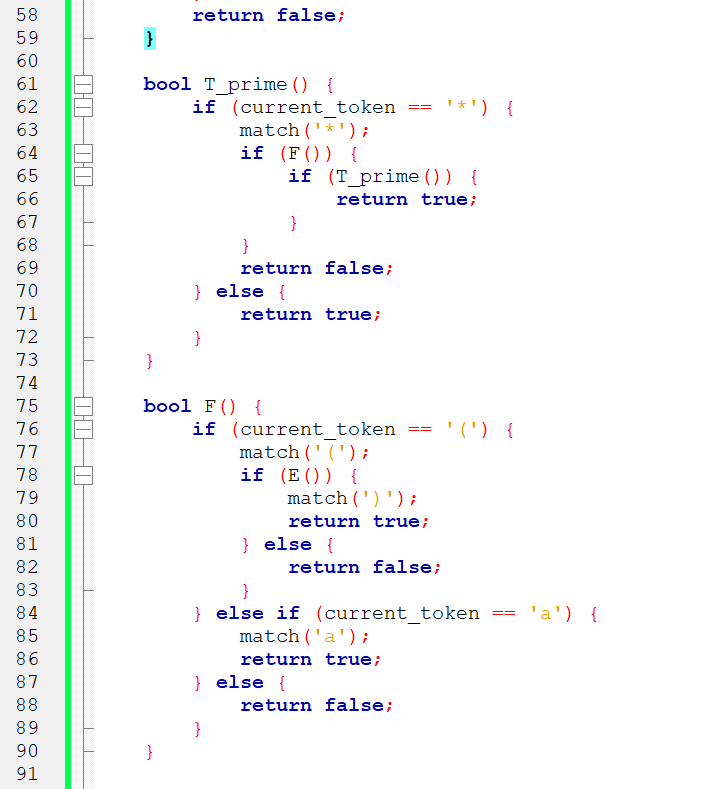
4. Start parsing from the start symbol of the grammar using the `parse` function.

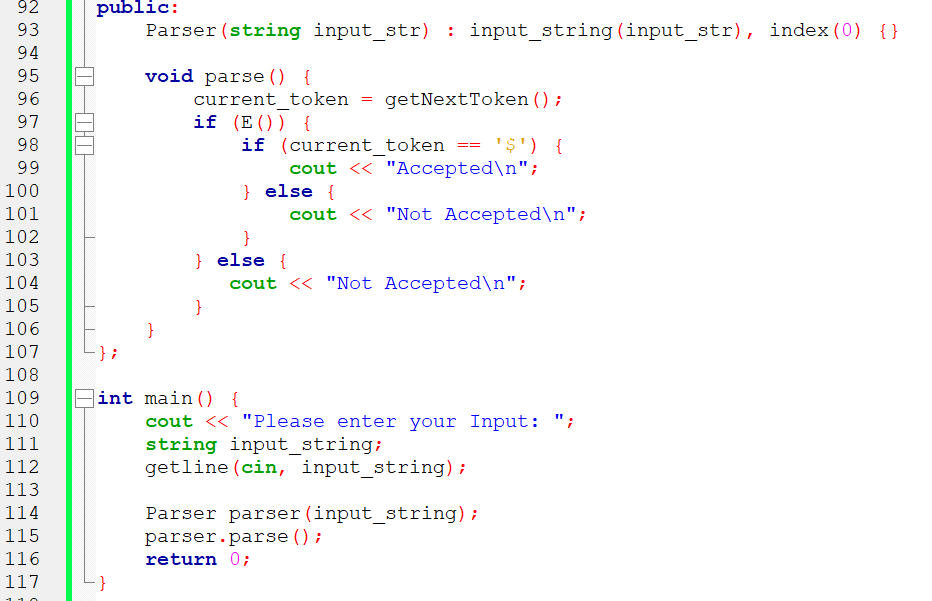
5. If the parsing is successful and the entire input is consumed, output "Accepted"; otherwise, output "Not Accepted".

**Code:**

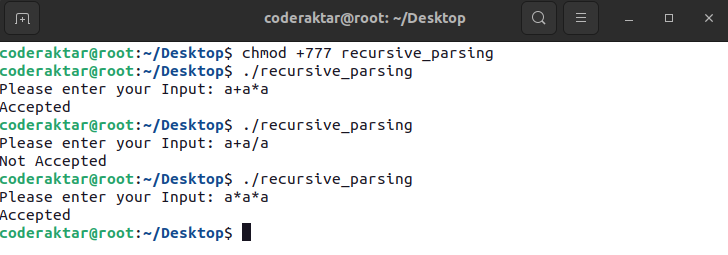








**Input and Output**



**Conclusion:** This program implements a recursive descent parser for the given grammar, allowing it to parse and determine whether a given input string is accepted by the grammar.