LEXER ANALYZER

Q1 Project Brief: Lexical Analyzer

Answer:

Objective:

Creating a program that performs lexical analysis on input text to identify and classify tokens based on defined rules.

Code Overview:

Language/Framework: C# using System and System.Collections.Generic namespaces.

Class Structure: Analyze class within the Analyzer namespace.

Functionalities:

loadTransitionTable: Reads transition rules from a file to set up the rules for token identification.

getNextState: Determines the next state based on the current state and input character.

isKeyword: Checks if a given token is a keyword or not.

Result:

Performs lexical analysis on the provided text.

Processes characters one by one, filtering out comments and identifying tokens based on predefined rules.

Recognizes identifiers, integers, floats, strings, and operators.

Builds a result string by replacing identified tokens with corresponding placeholders (<ID>, <INT>, <FLOAT>, <STR>, <OPR>).

Usage:

Input: Text input to be analyzed.

Transition Table: Rules for the analyzer loaded from a file (defaults to "matrix.txt").

Output: Resultant string after lexical analysis, replacing identified tokens with placeholders.

Functionality Notes:

The analyzer distinguishes between keywords, identifiers, integers, floats, strings, and operators based on the defined rules.

Comments in the input text are filtered out and not included in the output.

Identified tokens are replaced with their corresponding placeholders in the final output.