**ASSIGNMENT NO 4**



Submitted By :Nida Eman

Reg No: fa21-bcs-002

Submitted to : Sir Bilal Haider

Course: Compiler Construction

Date: 3/1/2025

**QS : Explain 2 core function of MiniCompiler?**

* Tokenize Function
* Optimizie function

**Tokenize Function:**

In OptiMIniCompiler , it will Tokenize function as part of the Lexical Analysis phase of a compiler. Its purpose is to scan the raw input source code and break it into manageable units called tokens.Tokens are the smallest meaningful elements in the source code, such as keywords (int) identifiers (`a`), operators (`=`), literals (`10`), and delimiters (`;`).

**Function Workflow:**

1. Input: Source code as a single string (e.g., `int a = 10;`).

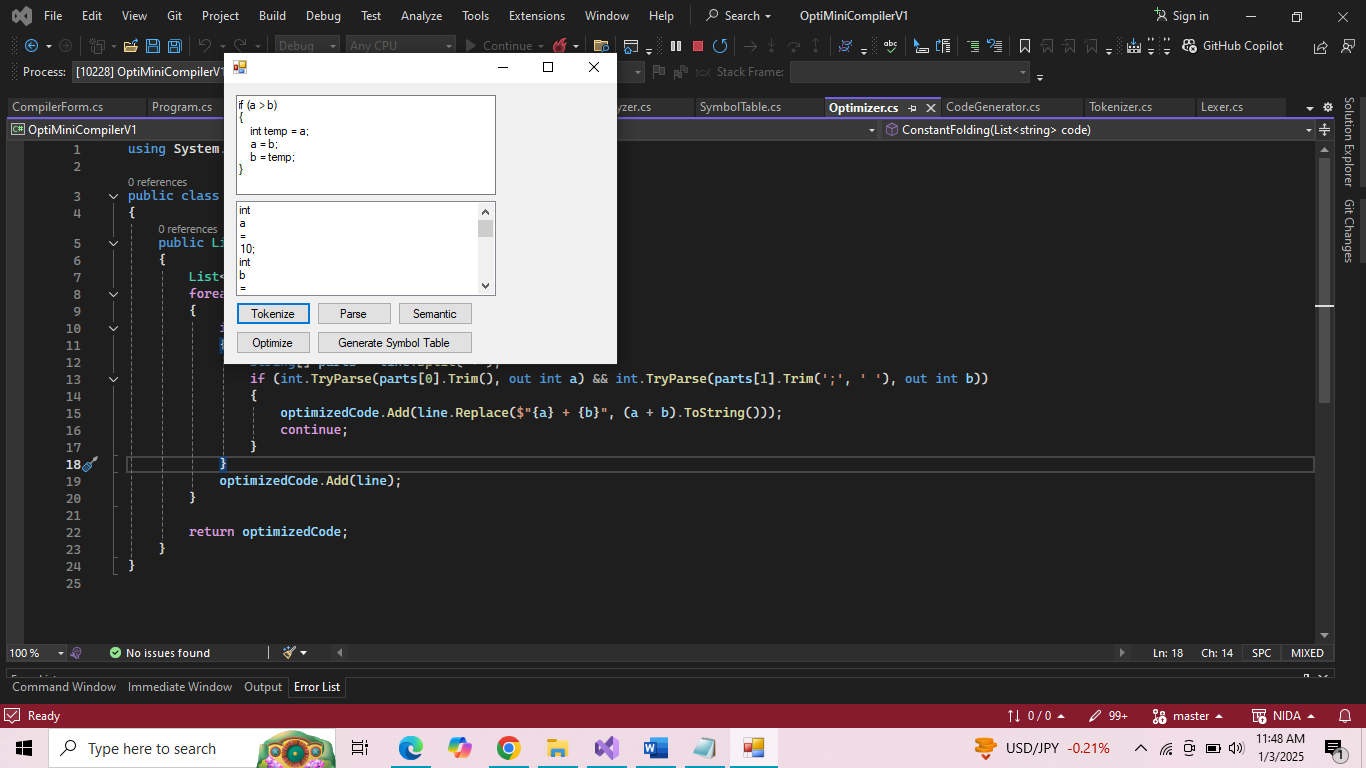
2. Process:

* The function uses a splitting mechanism (e.g., `string.Split()`) to divide the source code into smaller chunks based on spaces or other delimiters.
* Each chunk is checked to determine its type (keyword, identifier, operator, etc.).

3. Output:

- The function will returns each token for further processing.

Ouput :



**2) Optimize Function:**

**Purpose:**

The `Optimize` function belongs to the Optimization phase of a compiler. Its purpose is to improve the efficiency of the source code by simplifying expressions and eliminating redundant or unnecessary operations, while ensuring the behavior of the program remains unchanged.

**Optimization Techniques**:

* Constant Folding: Precomputing constant expressions (e.g., `2 + 3` becomes `5`).
* Dead Code Elimination: Removing unused or unreachable code.
* Simplification: Reducing complex expressions to simpler forms (e.g., `a = a + 0;` becomes `a;`).

**Function Workflow:**

* Input: Source code string.
* Process:
* Identify patterns of redundancy or inefficiency using techniques like string replacements or pattern matching.
* Apply transformations to simplify or remove unnecessary code.
* Output:
* The optimized source code string.

Ouput:

