

**B.TECH-CSE**

**SEMESTER VI**

**Principles of Compiler Design**

**COURSE CODE:**

**4CS501CC25**

**PRATICAL NO.: 2**

SUBMITTED BY:

**NAME: Paryusha Shah**

**ROLL NO.:22BCE234**

**BATCH: D2**

**Aim:**1. Recognize Keywords, Identifiers, and Numbers

2. Count the Number of Lines, Words, and Characters

3. Check Validity of an Identifier

4. Match Floating-Point and Integer Numbers

5. Remove Comments from C Code

6. Tokenize a Simple Arithmetic Expression

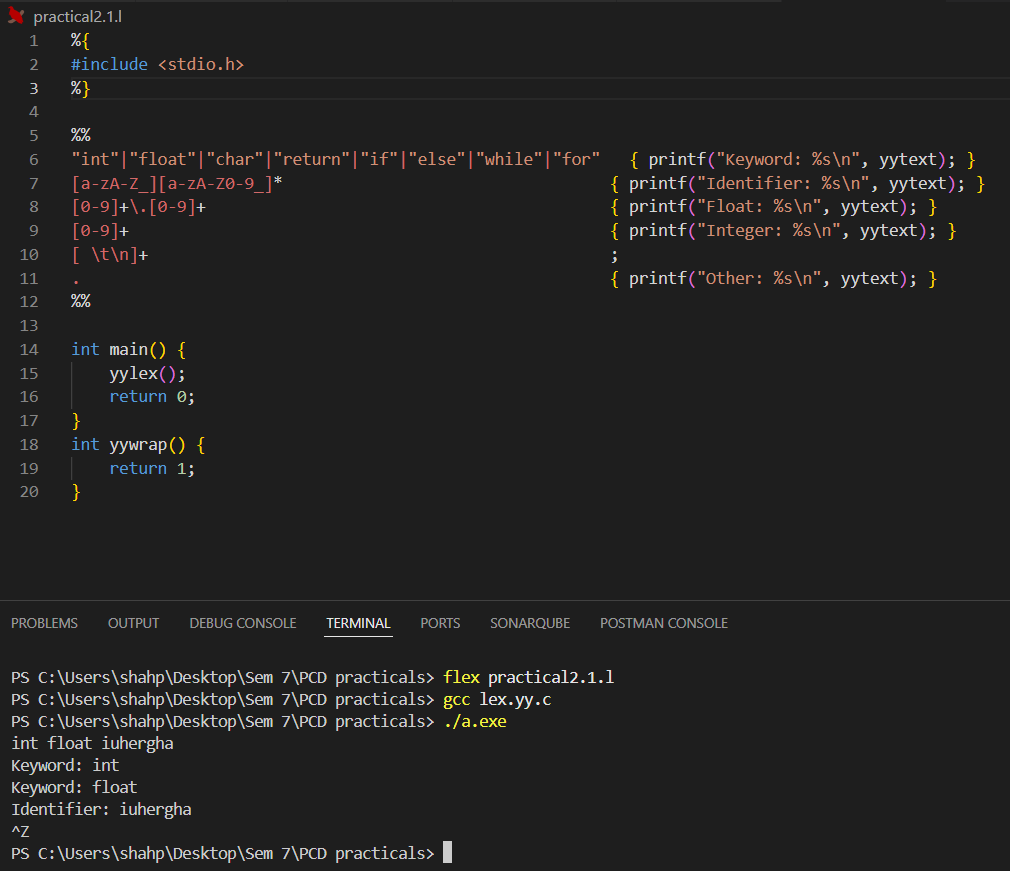
7. Detect Email Addresses

**1.  
Logic:**This program is designed to identify and classify basic elements (tokens) of a C-like programming language such as:

* Keywords (int, float, etc.)
* Identifiers (variable names)
* Numbers (integers and floats)
* Any other characters

**Description:**

* Uses regular expressions to match different types of tokens.
* Uses printf to print the token type and matched value.
* Whitespace is skipped.
* Non-matching characters are classified as Other.



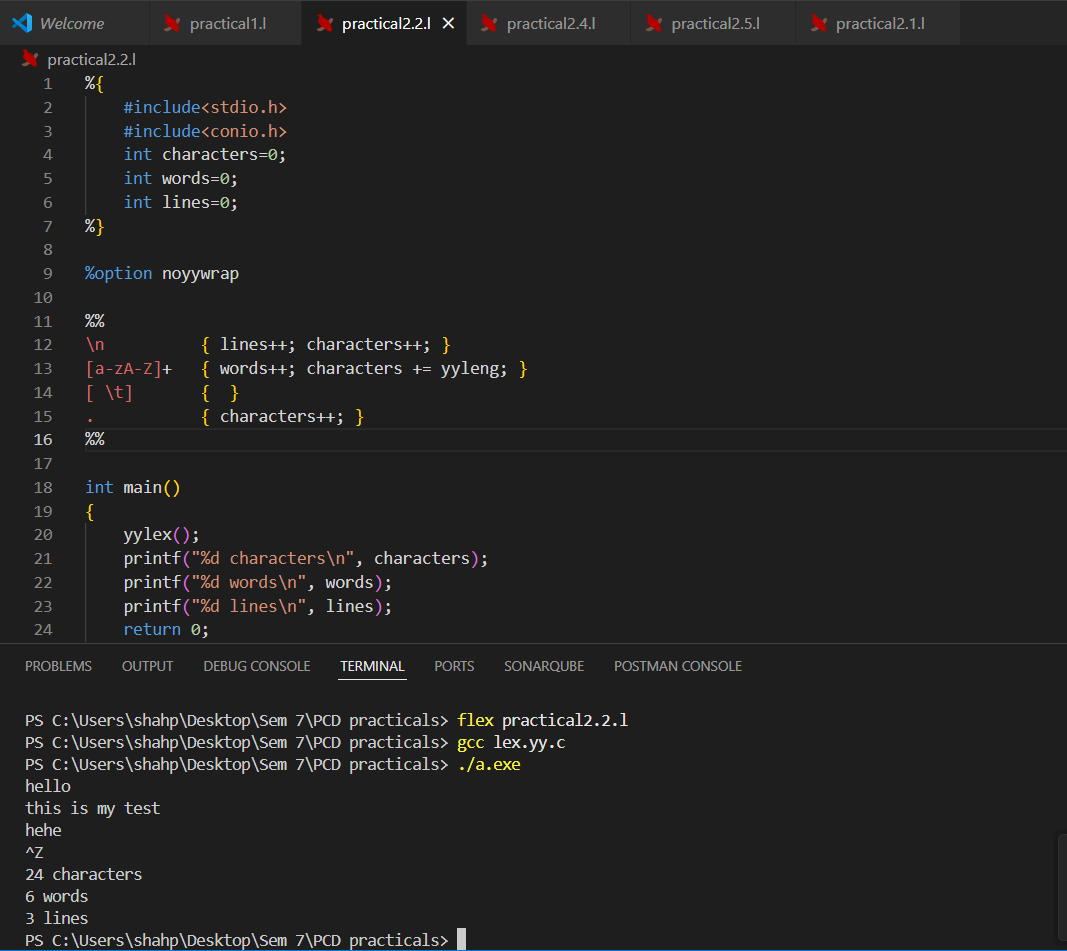
**2.** Count the Number of Lines, Words, and Characters

**Logic:**This program mimics the basic functionality of the Unix wc command:

* Counts the number of lines, words, and characters in the input.

**Description:**

* Increments lines on each newline (\n).
* Increments words for every non-whitespace word.
* Increments chars for all characters, including whitespace.
* Displays the final count after lexical analysis.

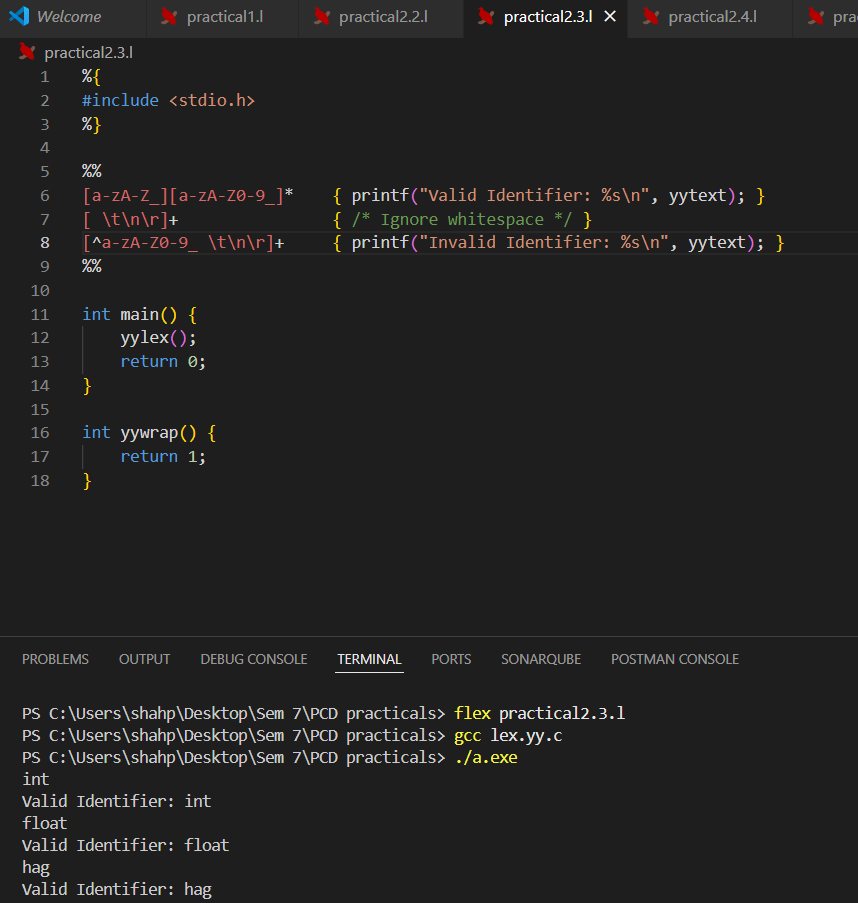
****

**3.**

**Logic:**This program checks whether an input string is a valid identifier.

**Description:**

* Valid identifiers start with a letter or underscore and can be followed by letters, digits, or underscores.
* Prints "Valid Identifier" or "Invalid Identifier" based on pattern matching.
* Useful for validating variable names in programming**.**

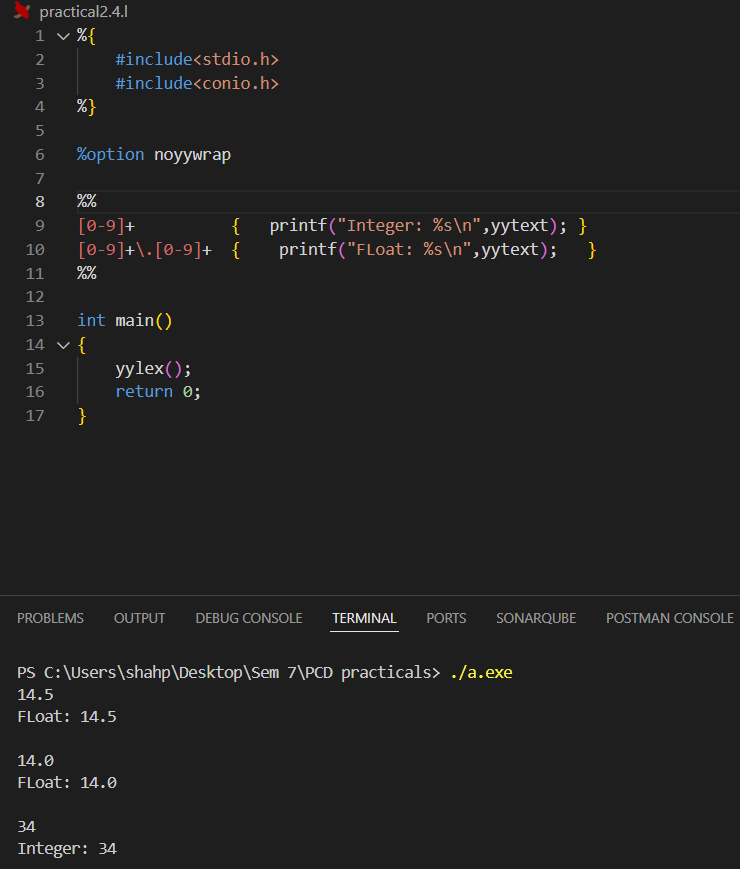
****

**4.**

**Logic:**This program classifies numeric values as integers or floating-point numbers, including scientific notation.

**Description:**

* Recognizes floats with optional exponent part (e.g., 3.14, 1.2e10).
* Recognizes integers without decimal points.
* Ignores other characters.
* Helpful for analyzing numeric inputs in code.

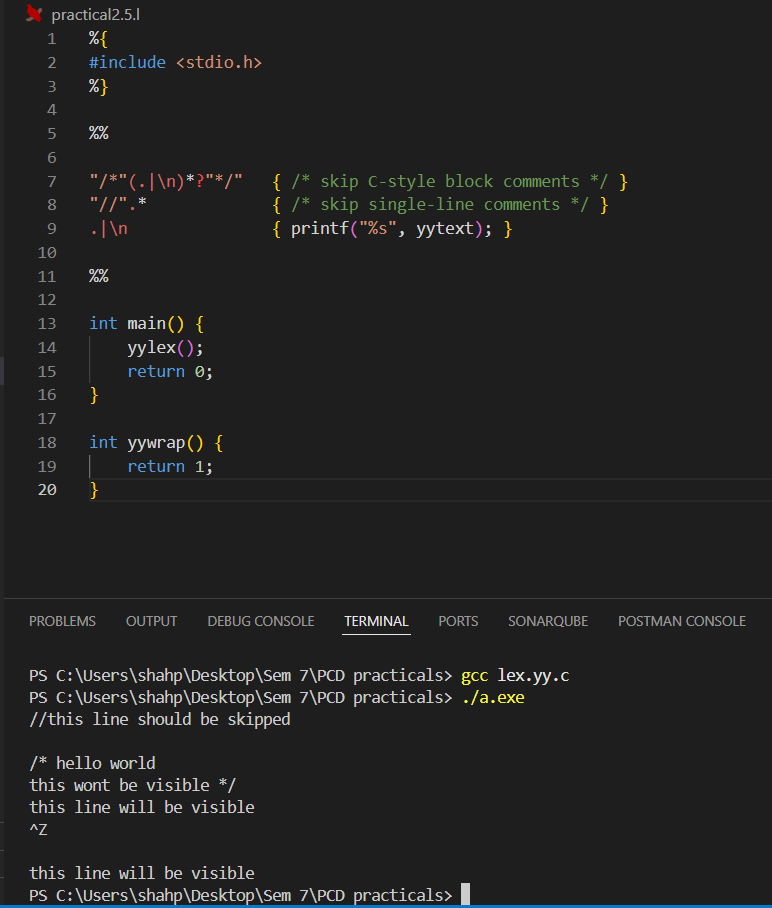
****

**5.**

**Logic:**This program removes single-line (//) and multi-line (/\* ... \*/) comments from source code.

**Description:**

* Matches and skips both block and line comments.
* Prints all other input characters unchanged.
* Useful for preprocessing code to extract logic without comments.

**w**

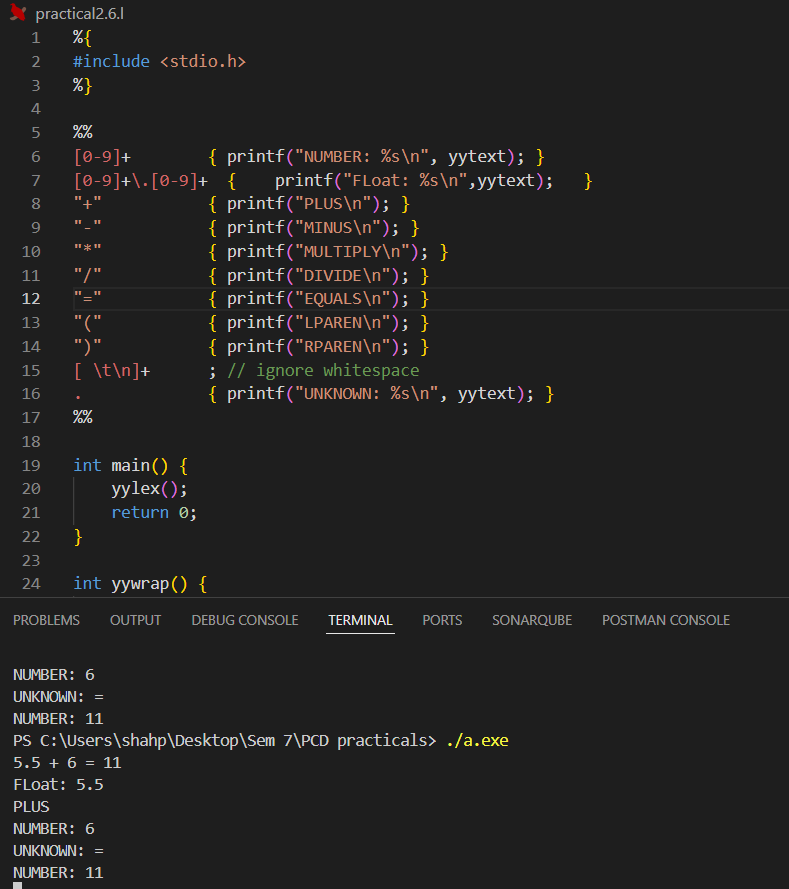
**6.**

**Logic:**This program tokenizes arithmetic expressions into:

* Numbers
* Operators (+, -, \*, /)
* Parentheses

**Description:**

* Each matched token is printed with its type (e.g., PLUS, MINUS).
* Whitespace is ignored.
* Unknown characters are flagged.
* Ideal for parsing simple math expressions.

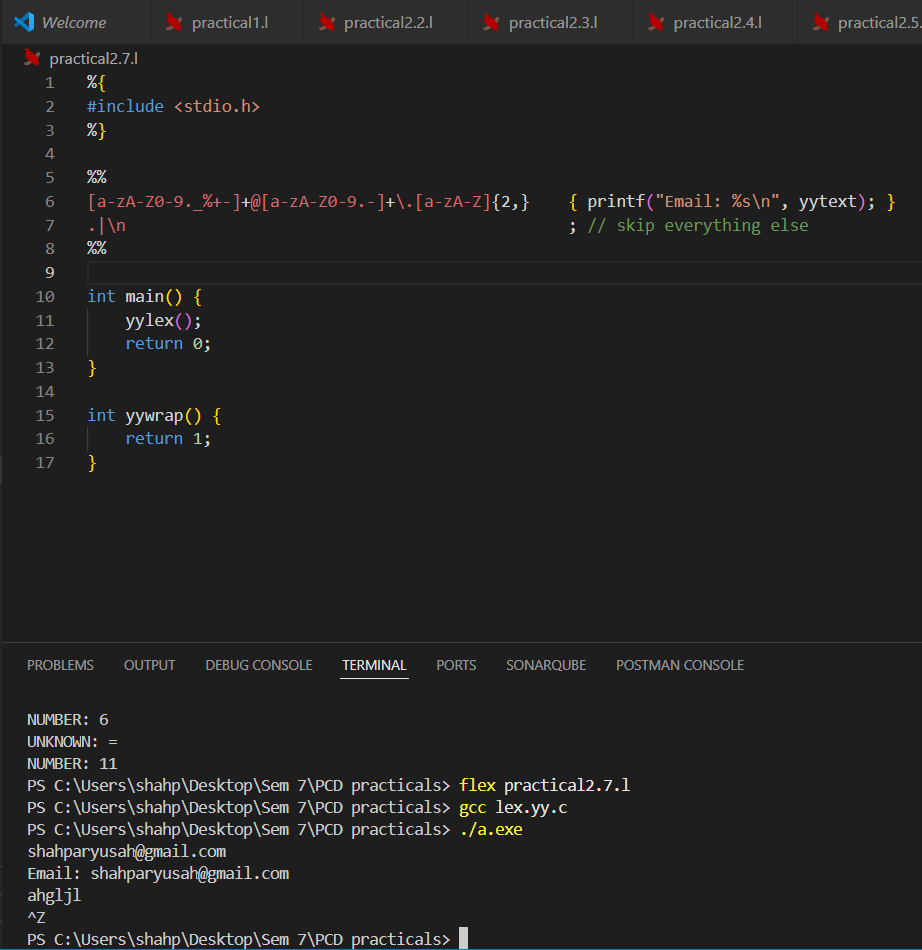
**w**

**7.**

**Logic:**This program identifies and prints valid email addresses from input text.

**Description:**

* Uses a regular expression to match standard email formats.
* Skips any non-matching input.
* Useful for scanning documents or logs for email extraction.

****