**AI -lab-1**

**Name : usman jawad**

**Submit to : Sir Rasikh Ali**

**Task: Dynamic calculator**

**Explanation of the code:**

* **Main Function (main)**
* This is where the program starts.
* It shows a welcome message and a sample expression.
* It runs a loop that keeps asking the user for input until they type "quit".
* For each input, it calls the calculate() function and prints the result.
* **Calculate Function (calculate)**
* First, it removes all spaces from the input so it’s easier to process.
* Then it converts the expression into a list of tokens (numbers, operators like +, -, \*, /, and parentheses).
* Example:  
  Input: "1 + (2 \* 3)"  
  Tokens: [1.0, '+', '(', 2.0, '\*', 3.0, ')']
* After tokenizing, it sends the tokens to the parse function to compute the result.
* **Parsing with Correct Order (Math Rules)**The program uses three levels of parsing to follow math rules:
* parse\_expr() → Handles addition (+) and subtraction (-)
* parse\_term() → Handles multiplication (\*) and division (/)
* parse\_factor() → Handles numbers and parentheses ( )

This ensures that:

* \* and / are done before + and -
* Anything inside ( ) is calculated first
* **Handling Special Cases**
* If the user types a negative number like -5, the code handles it correctly.
* If there's a division by zero (like 5 / 0), it shows an error: *"Cannot divide by zero"*.
* If parentheses are missing or there's an invalid symbol, it shows a clear error.

It follows the rules of mathematics such as:

* Operator precedence (e.g., multiplication and division come before addition and subtraction)
* Parentheses first (solve inside brackets first)
* Left-to-right evaluation for same-level operators

**Screenshoot:**

