**LAB terminal**

**Submitted By:**

Alveena Ayesha

FA21-BCS-014

**Course Instructor:**

Mr. Syed Bilal Bukhari

**Course:**

CC

**Date:**

3ND January, 2025

**Q5**

**explain functions that perfome semantic analyisis**

### Functions That Perform Semantic Analysis

In a typical mini compiler, these functions might be implemented for semantic analysis:

1. **Symbol Table Management**
   * **Purpose**: Tracks variable declarations, types, and scopes.
   * **Functionality**:
     + Insert new symbols (variables, functions) when declared.
     + Lookup symbols to ensure they are used after being declared.
     + Ensure no redeclaration of identifiers in the same scope.
2. **Type Checking**
   * **Purpose**: Ensures operations are performed on compatible data types.
   * **Functionality**:
     + Verify that expressions like int x = "text"; are invalid.
     + Check type compatibility in operations (e.g., int + float).
3. **Scope Checking**
   * **Purpose**: Enforces rules for variable scope.
   * **Functionality**:
     + Ensure a variable is used only within its declared scope.
     + Identify and report errors like accessing a variable outside its scope.
4. **Function Verification**
   * **Purpose**: Checks correctness of function calls and definitions.
   * **Functionality**:
     + Verify argument types and counts in function calls match their definitions.
     + Ensure functions return values of the declared return type.
5. **Control Flow Analysis**
   * **Purpose**: Ensures control structures are used correctly.
   * **Functionality**:
     + Check break and continue statements are inside loops.
     + Ensure functions have valid return statements in all control paths.
6. **Array Bounds Checking (Optional)**
   * **Purpose**: Detects out-of-bounds array accesses during compilation.
   * **Functionality**:
     + Ensure array indices are within declared limits.

.