CPL Assignment 3: Implement a symbol table handling nested scopes

A compiler records the variable names used in the source program along with its attributes and stores it in a data structure called a symbol table.

Write a code to support implementation of a symbol table with nested scopes using a hash table. Maintain a new symbol table for every scope and link it with the surrounding scope. Provide functionality to insert and lookup for a variable in the current scope hash table. If the variable entry does not exist in the current scope hash table, look up the value in the surrounding scope.

Below are the functionalities that need to be implemented.

begin	**Initialize Scope** - increments the current level and creates a new symbol table for that level. It links the new table to the previous level's table and updates the current level pointer used by lookup and insert functions.
assign a 1	**Insert** the value of variable a in the current scope with value 1
print b	**Lookup** the value of b in the current scope. If it does not exist in the current scope look it up in the surrounding scope
end	**Finalize Scope** changes the current level pointer so that it points to the table for the scope surrounding the current level

Sample code and its output

```
begin
        assign a 1
        assign b 2
        begin
                assign a 3
                assign c 4
                print b                                 => prints value 2
                begin
                        assign c 5
                        print b                         => prints value 2
                        print a                         => prints value 3
                        print c                         => prints value 5
                end
                print a                                 => prints value 3
                print c                                 => prints value 4
        end
        print a                                         => prints value 1
end
```