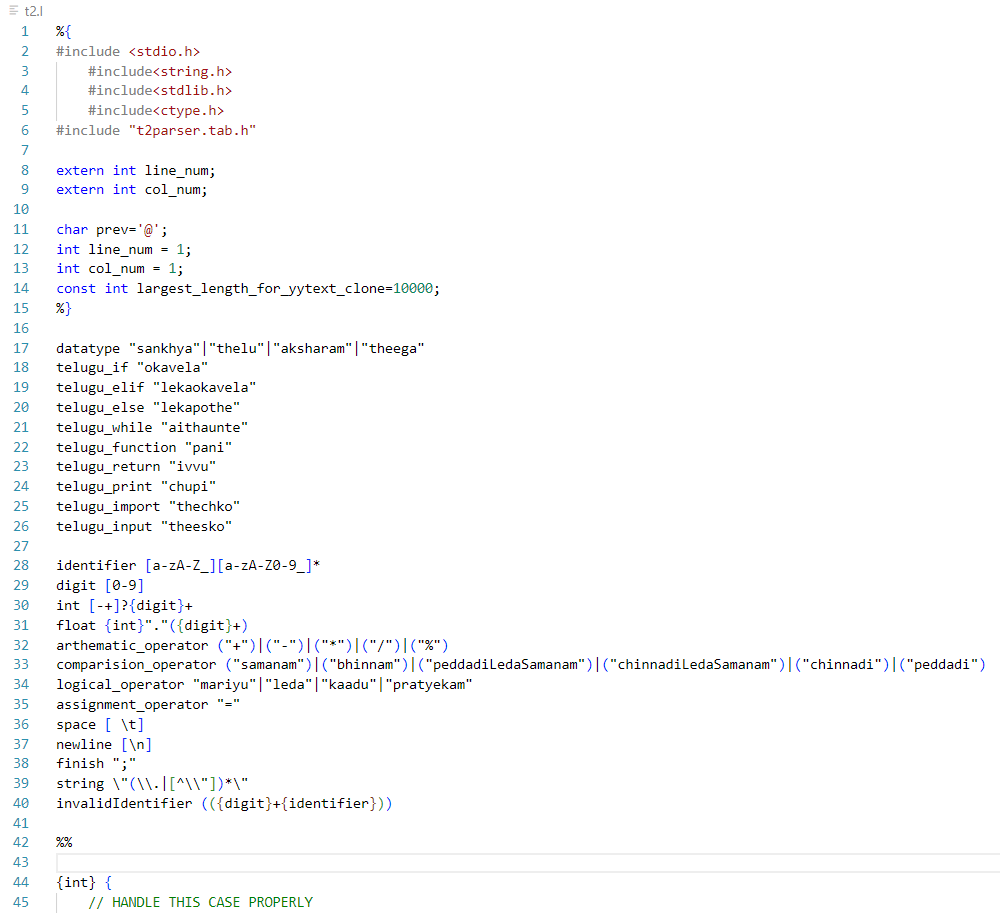
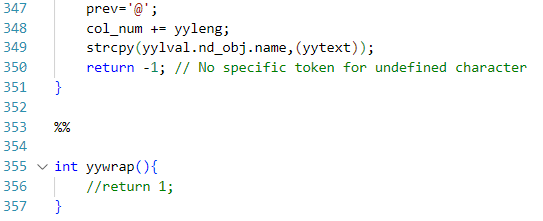
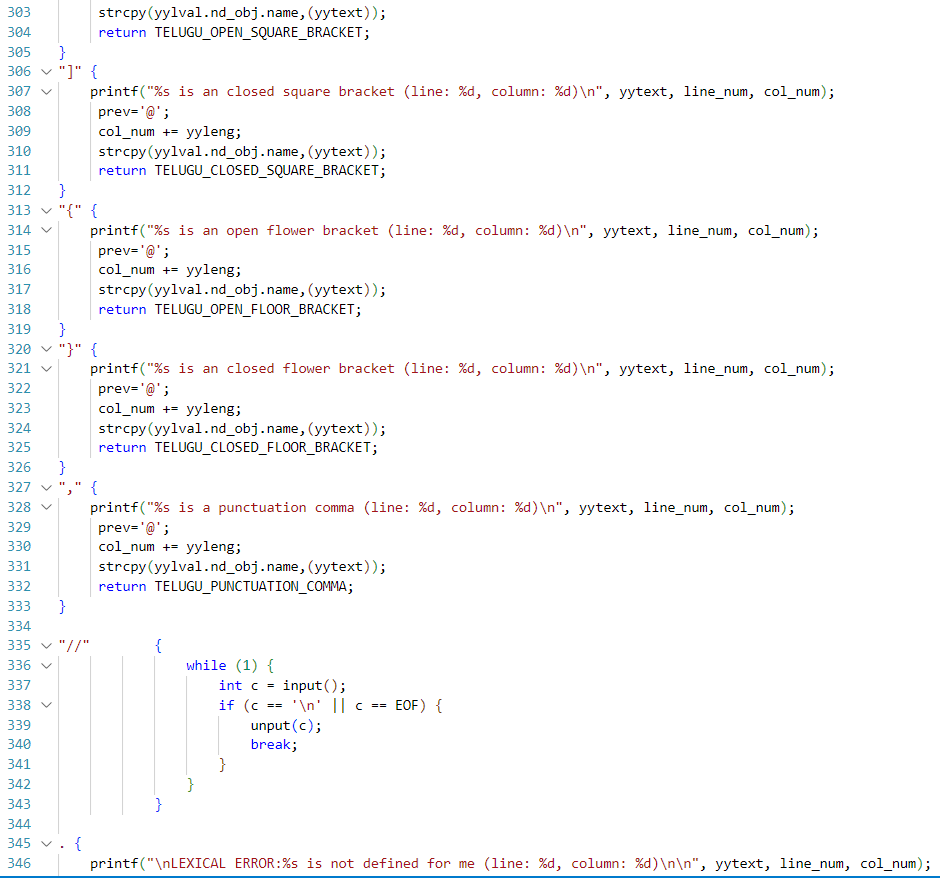
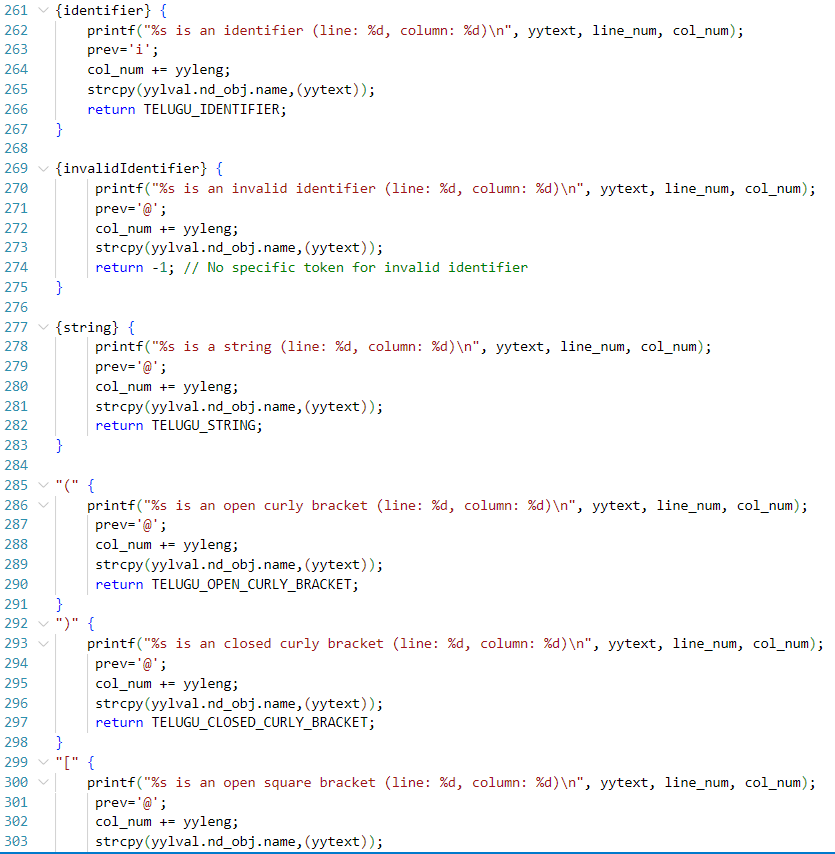
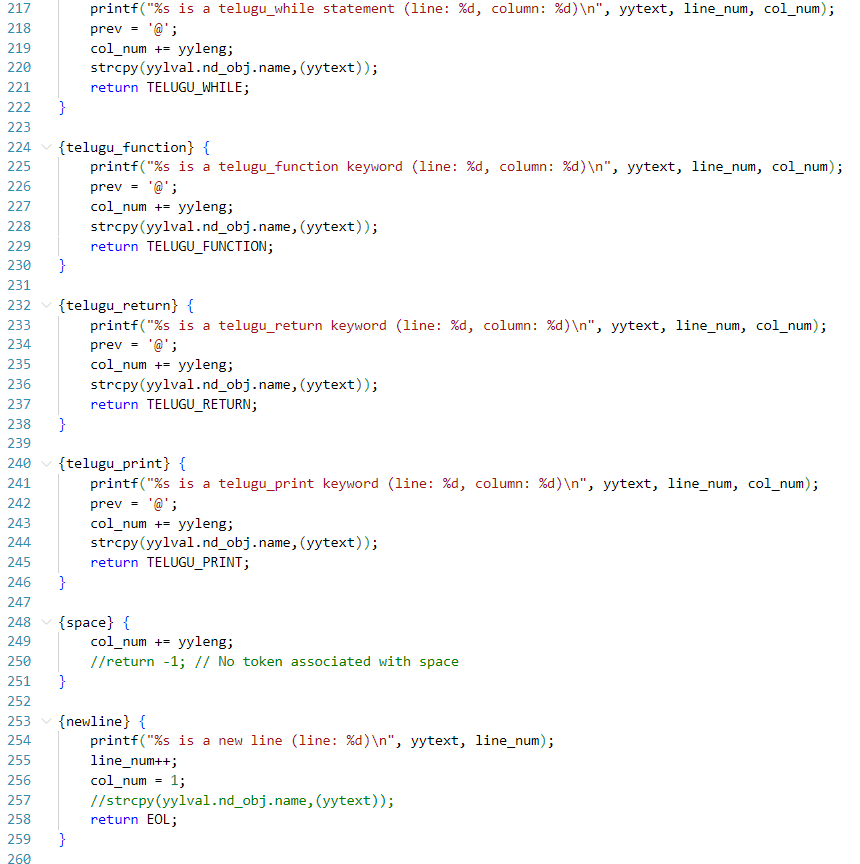
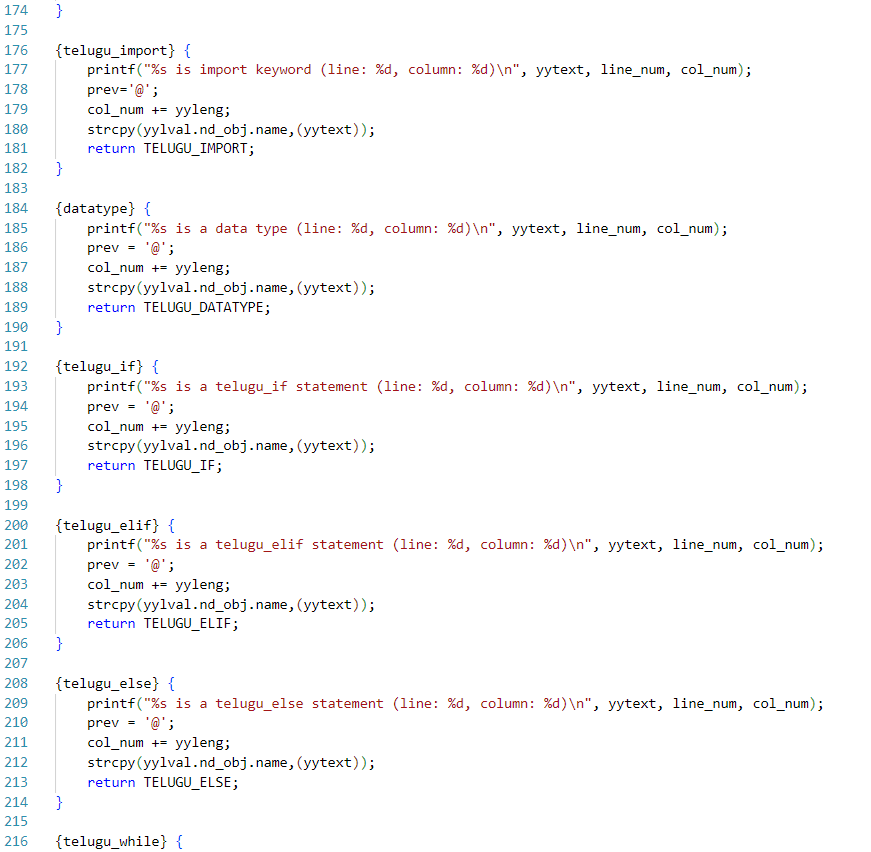
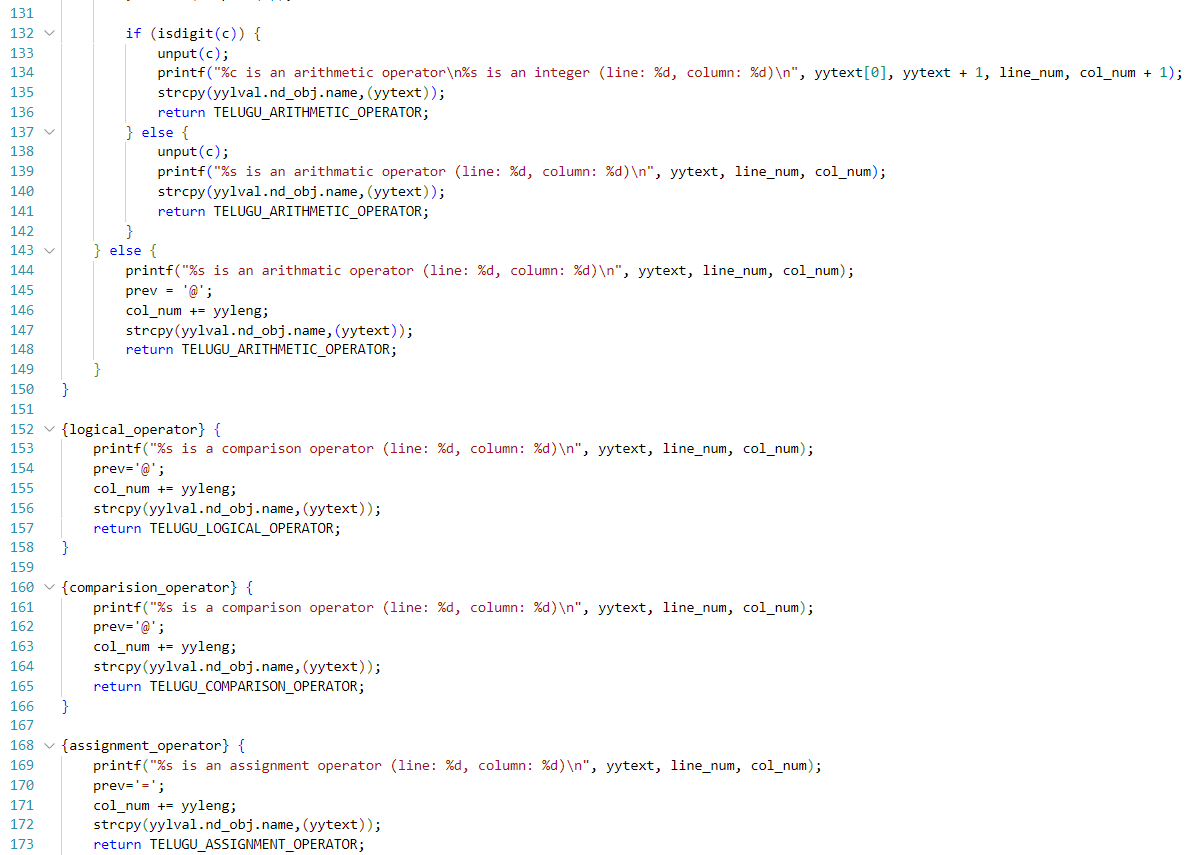
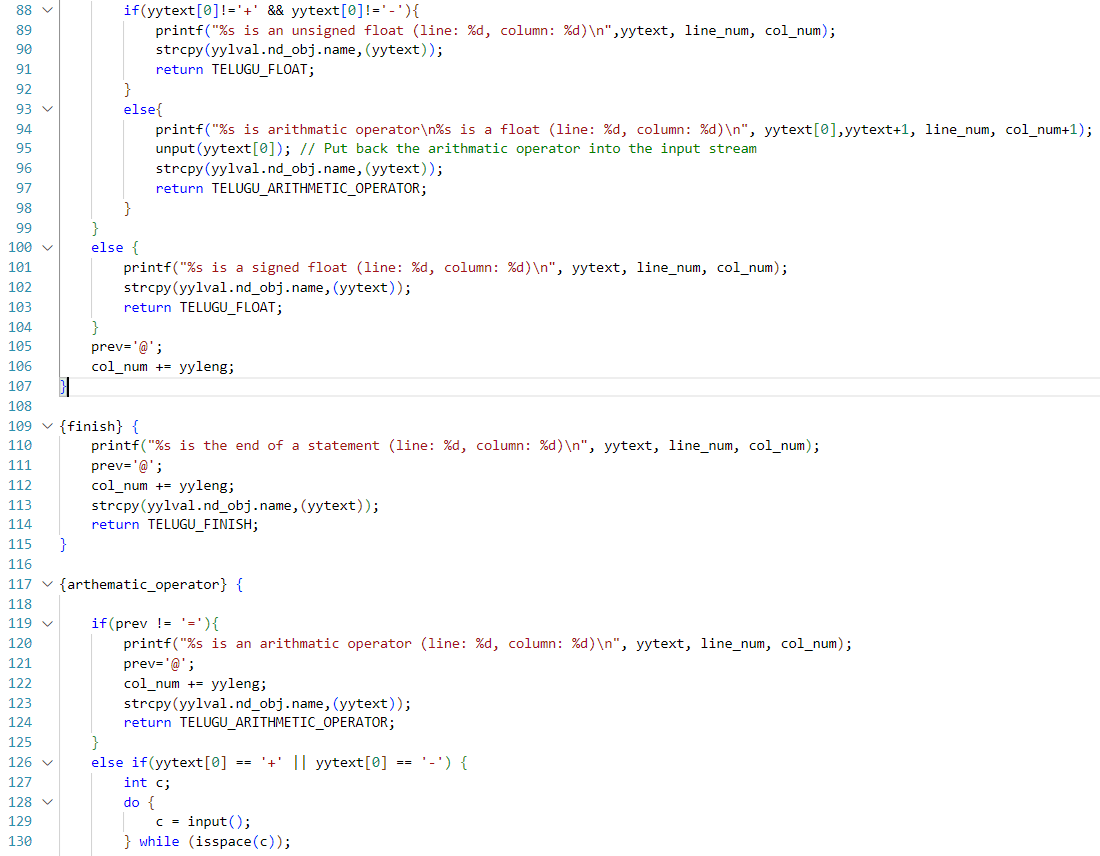
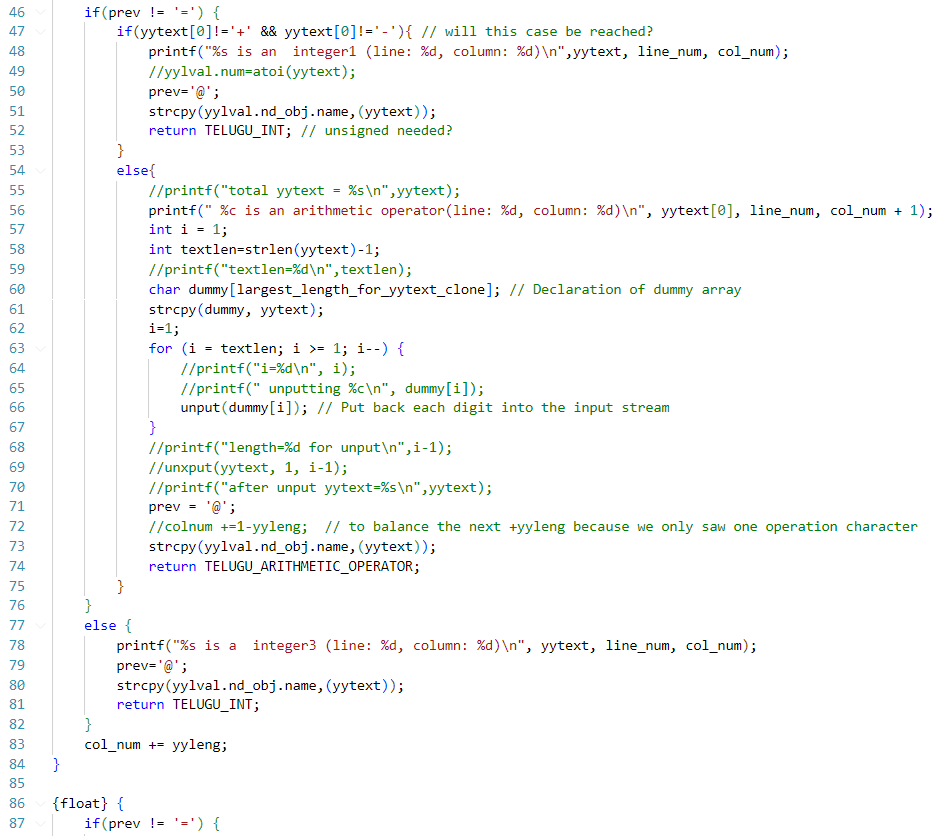
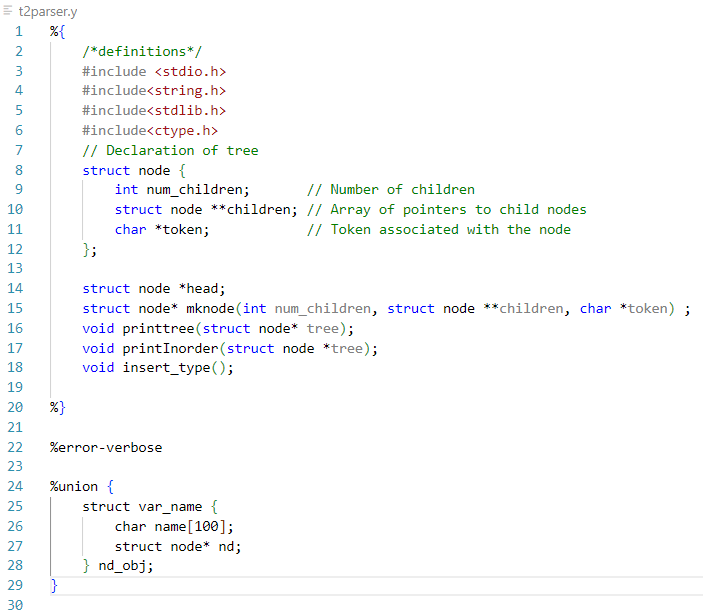
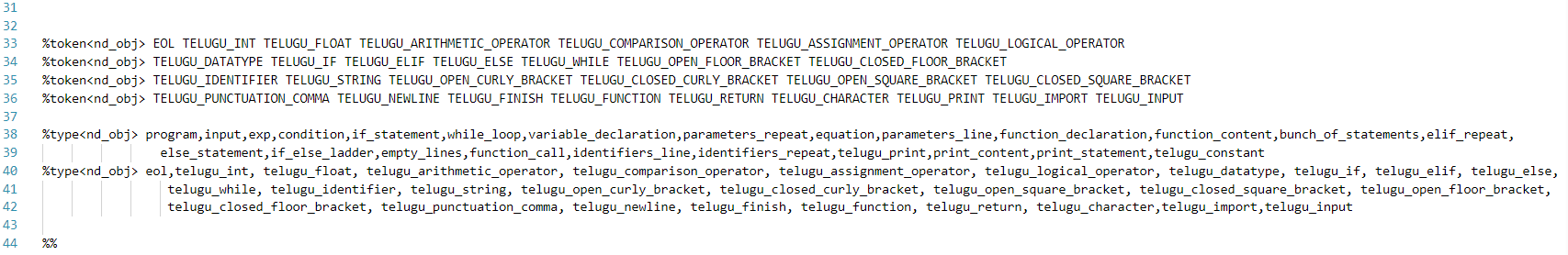
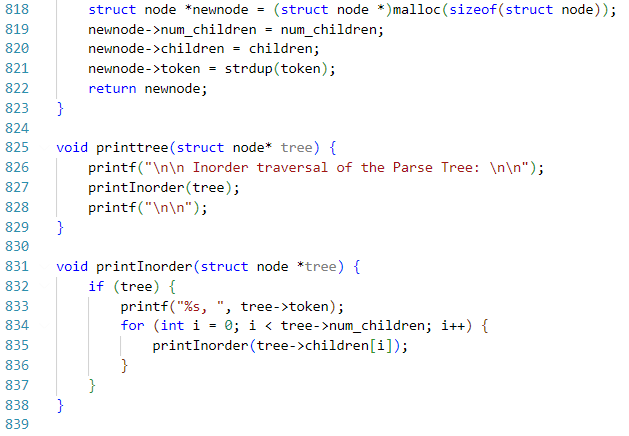
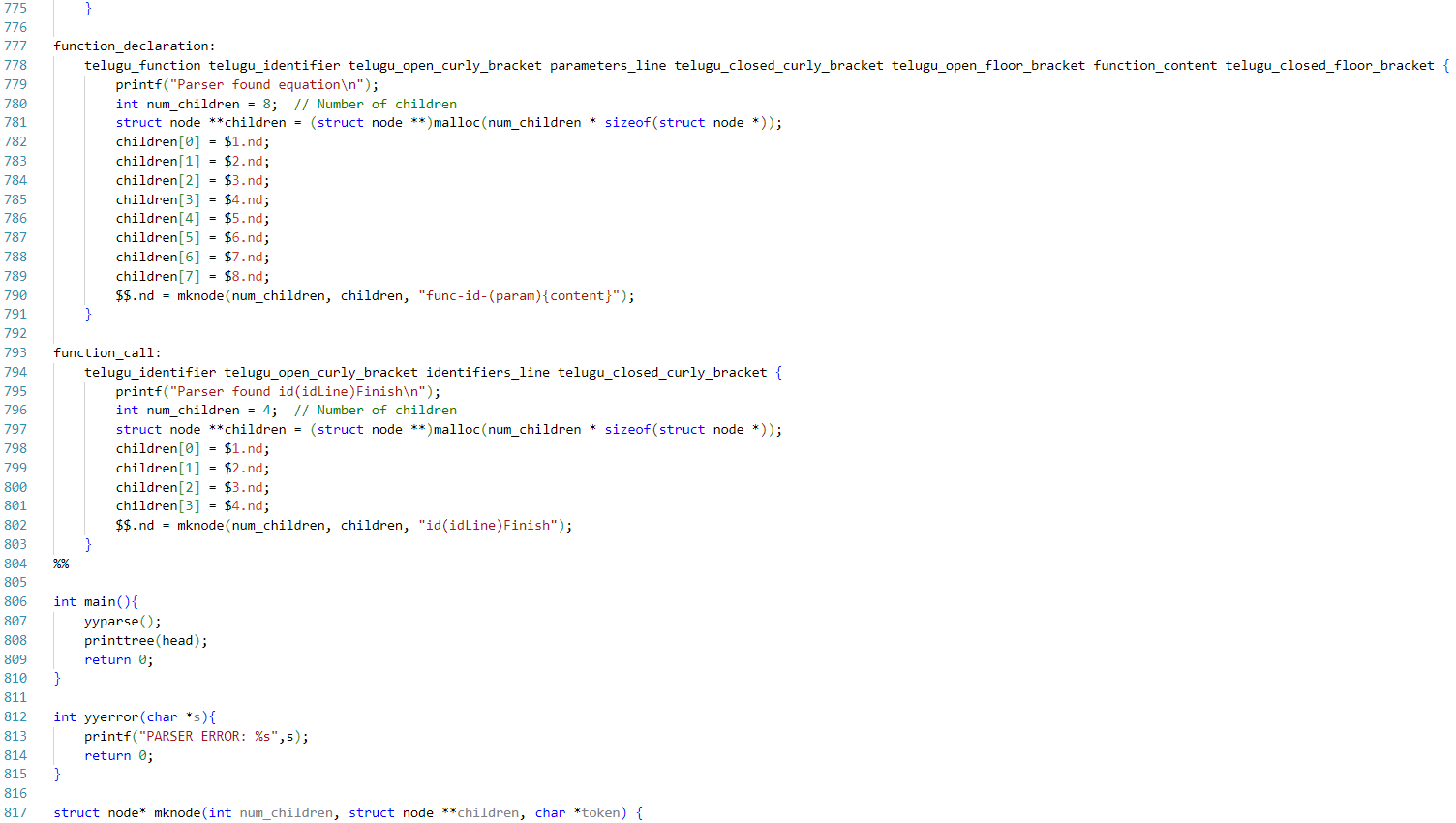
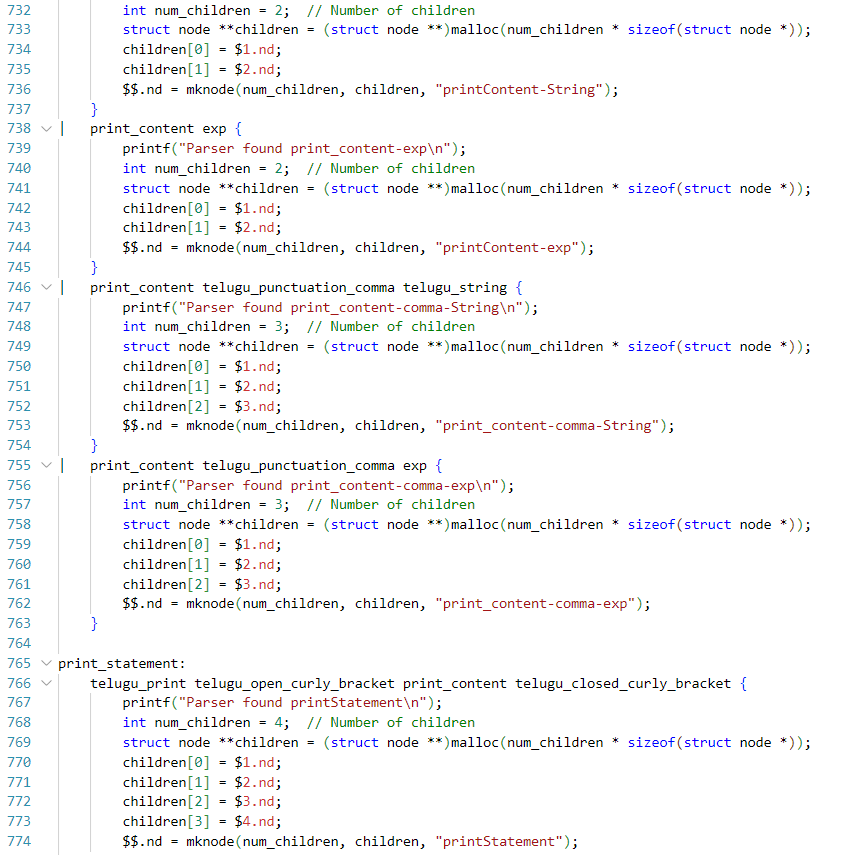
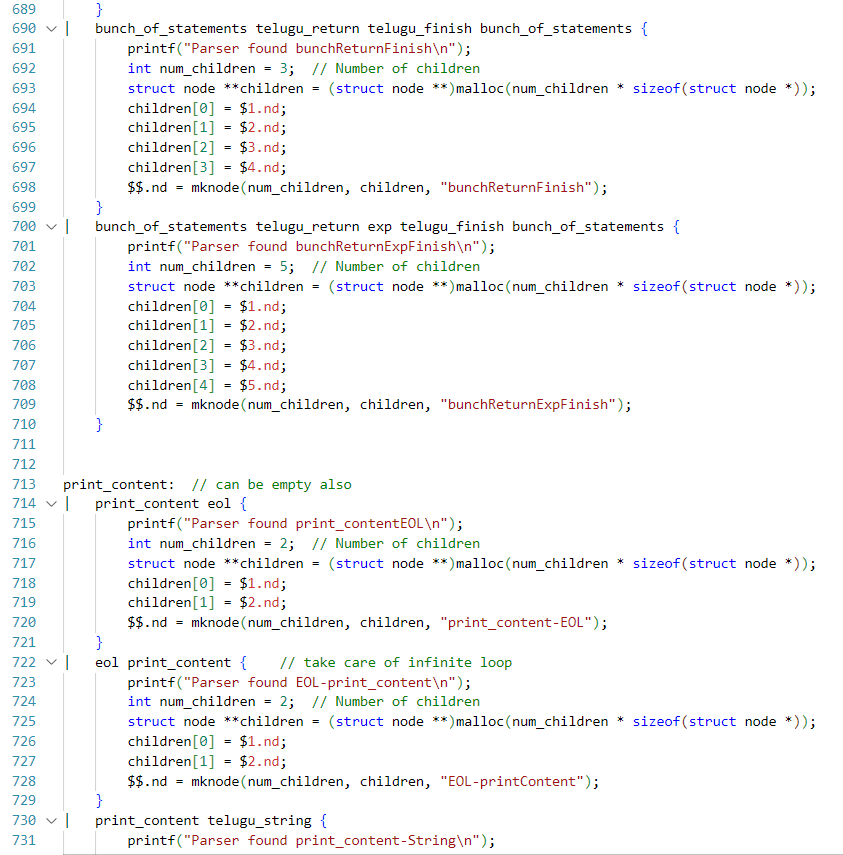
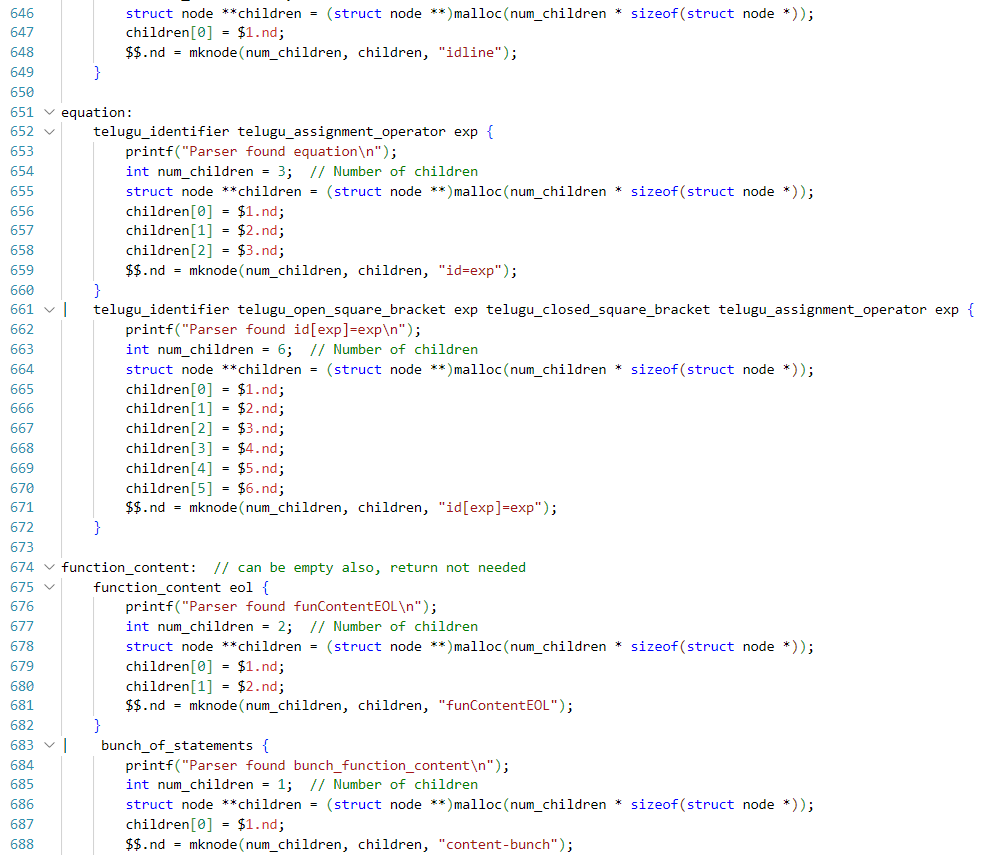
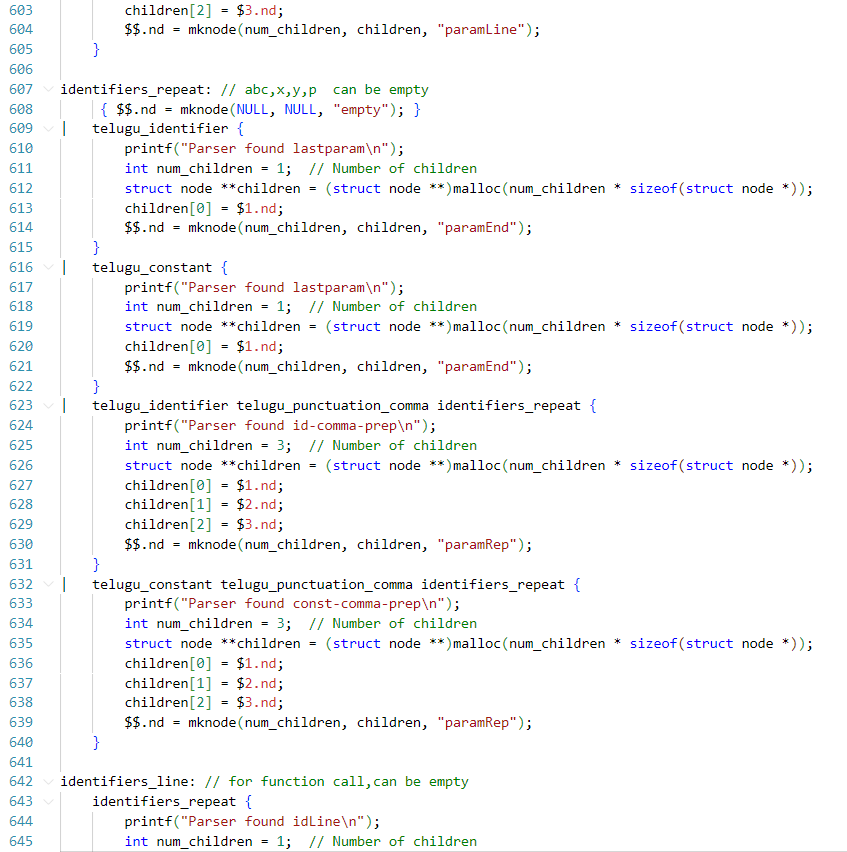
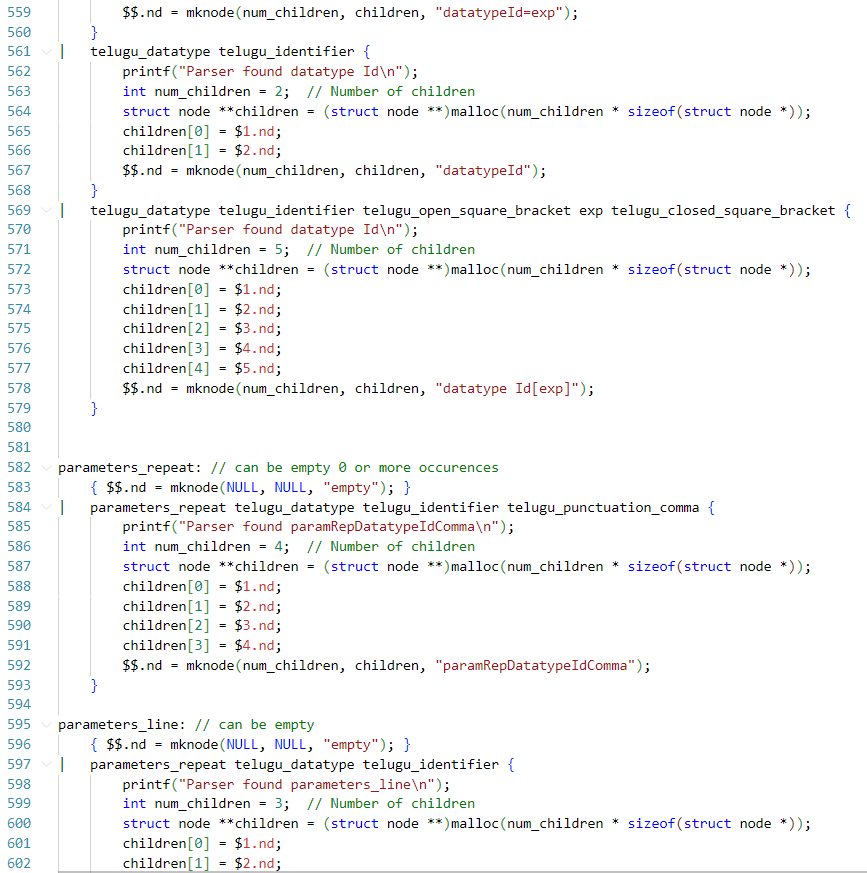
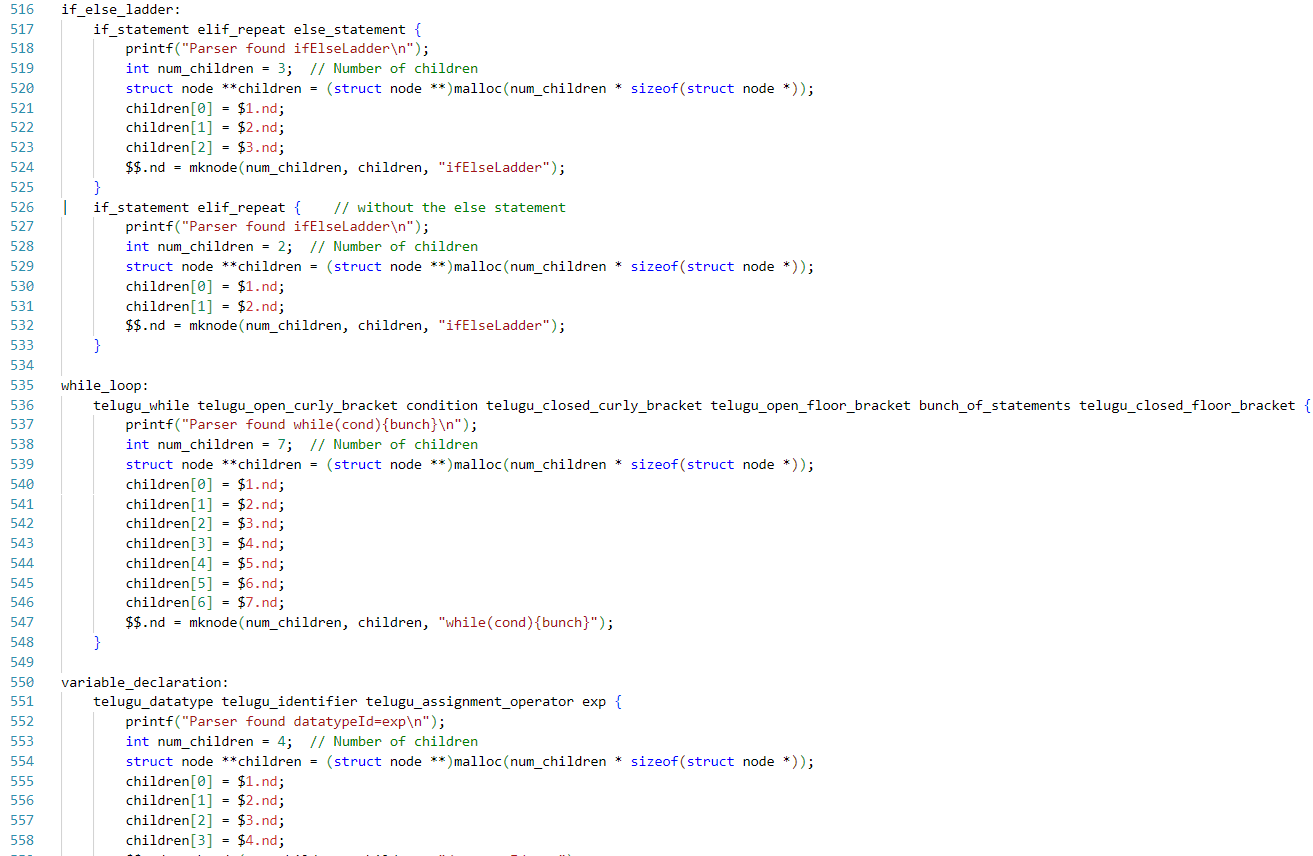
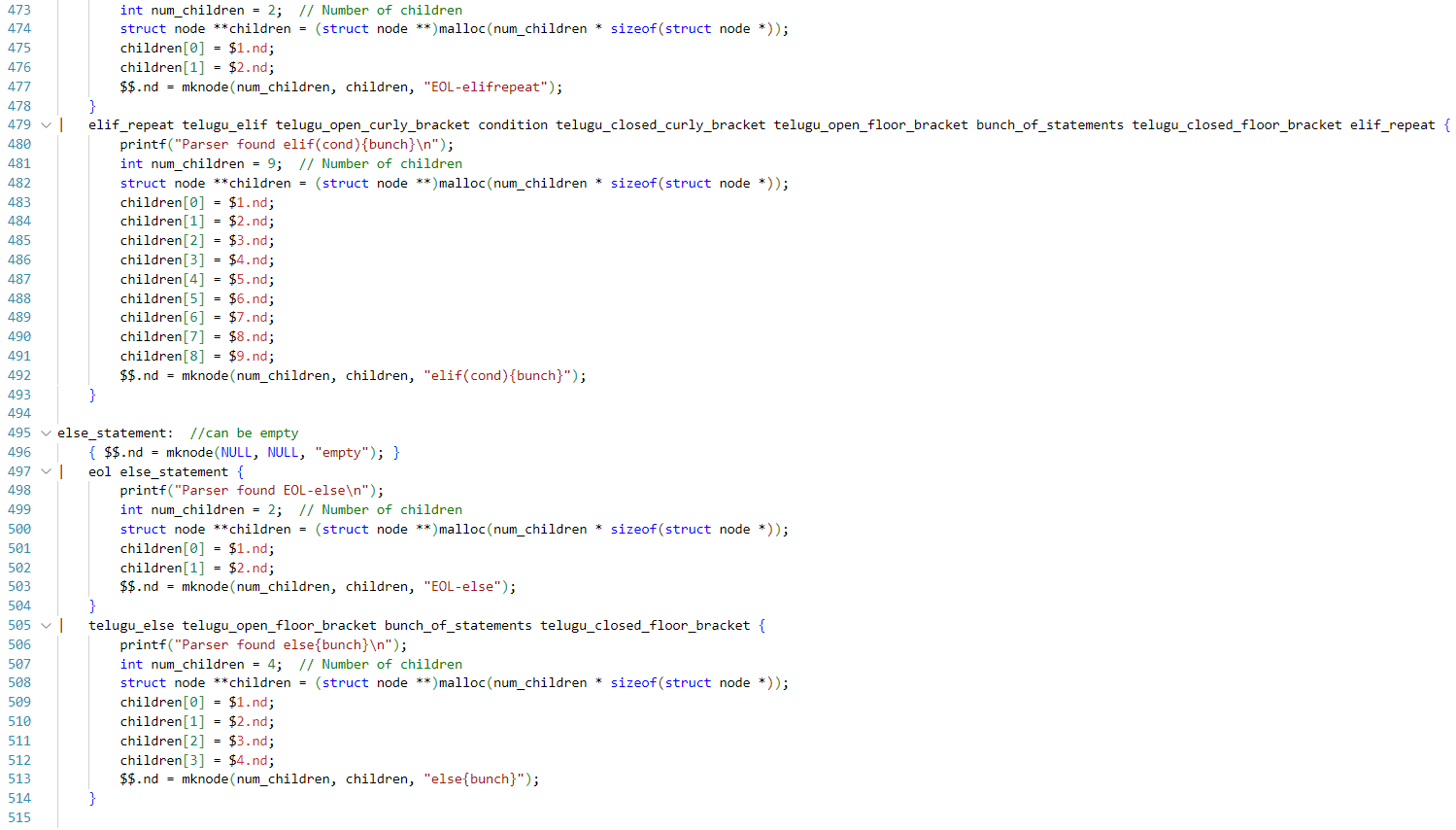
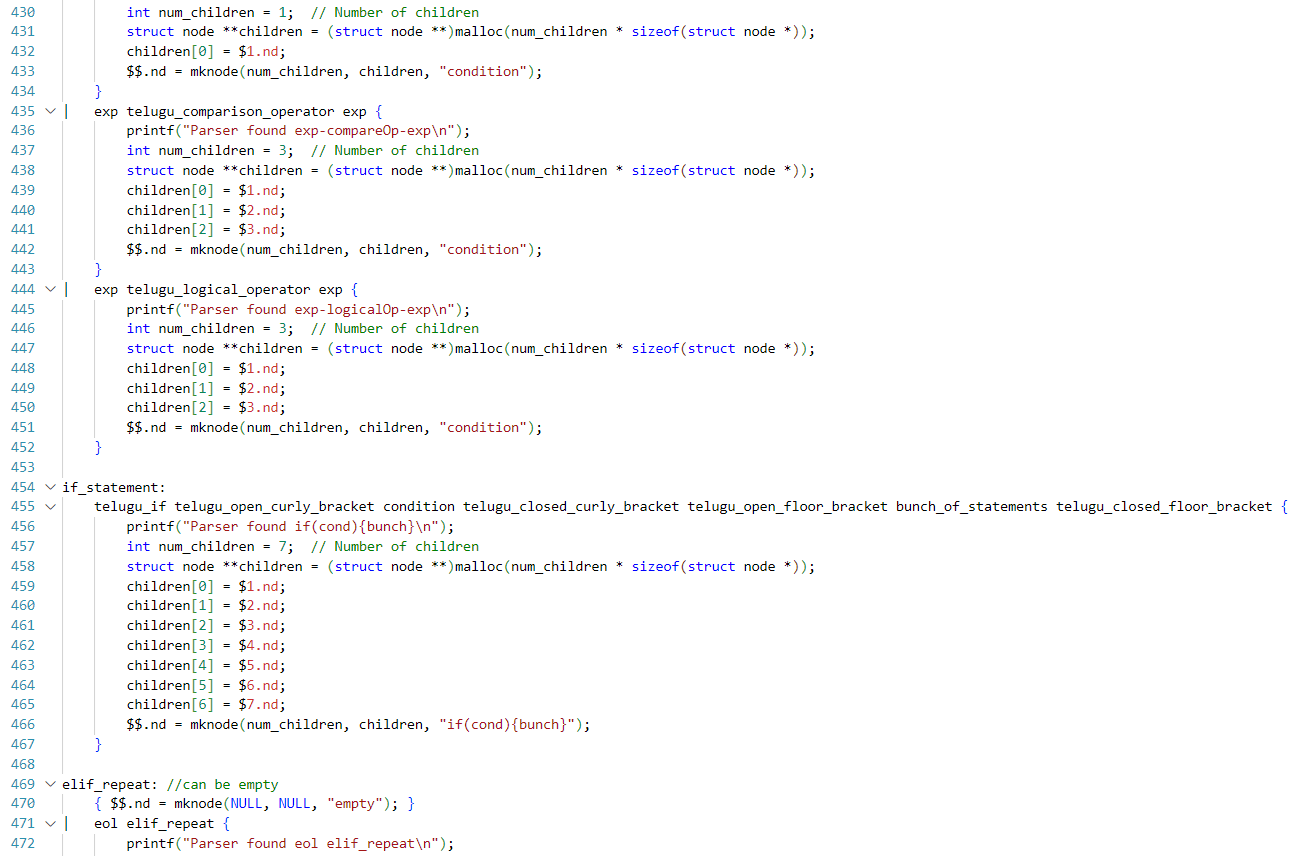
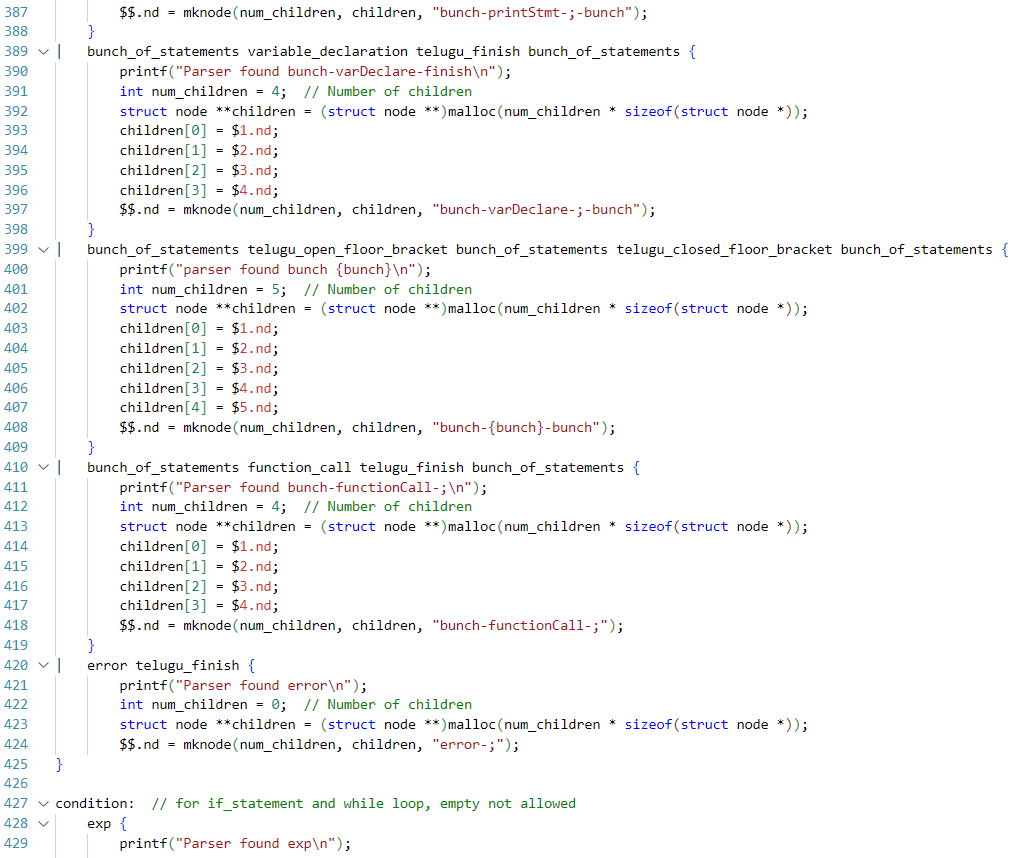
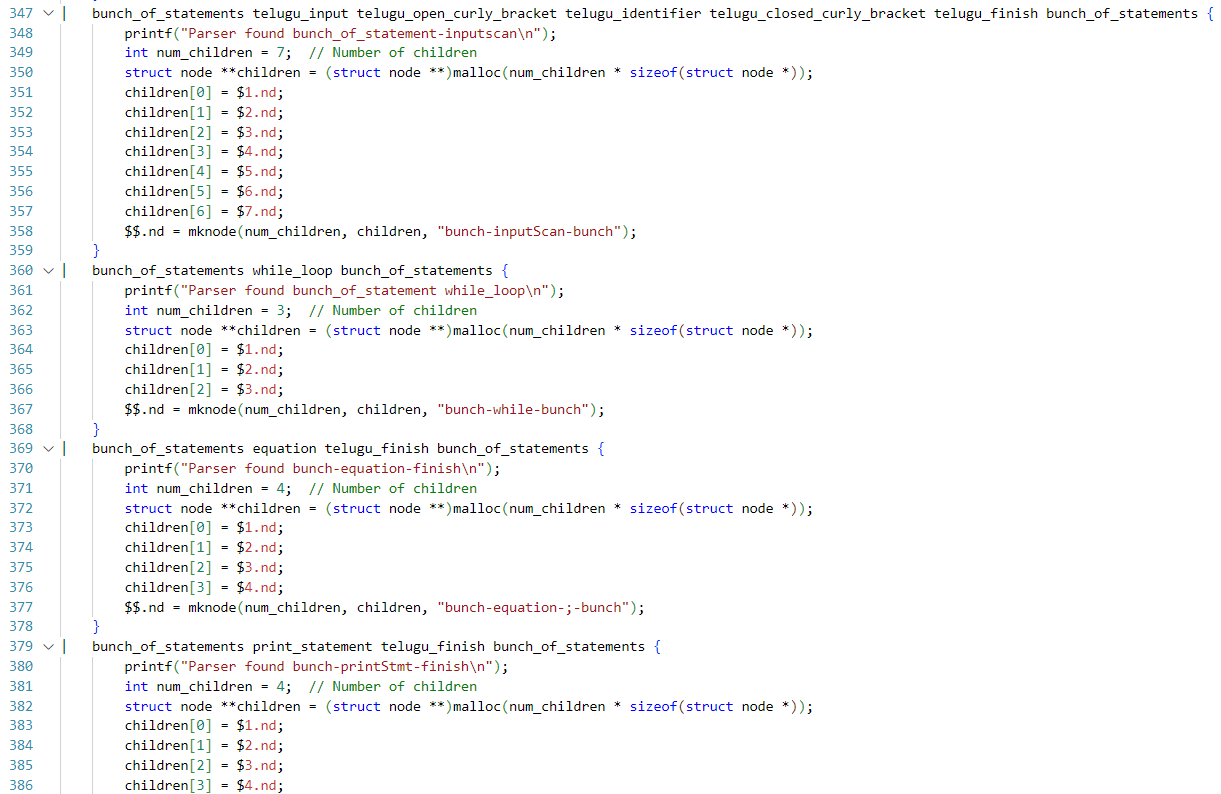
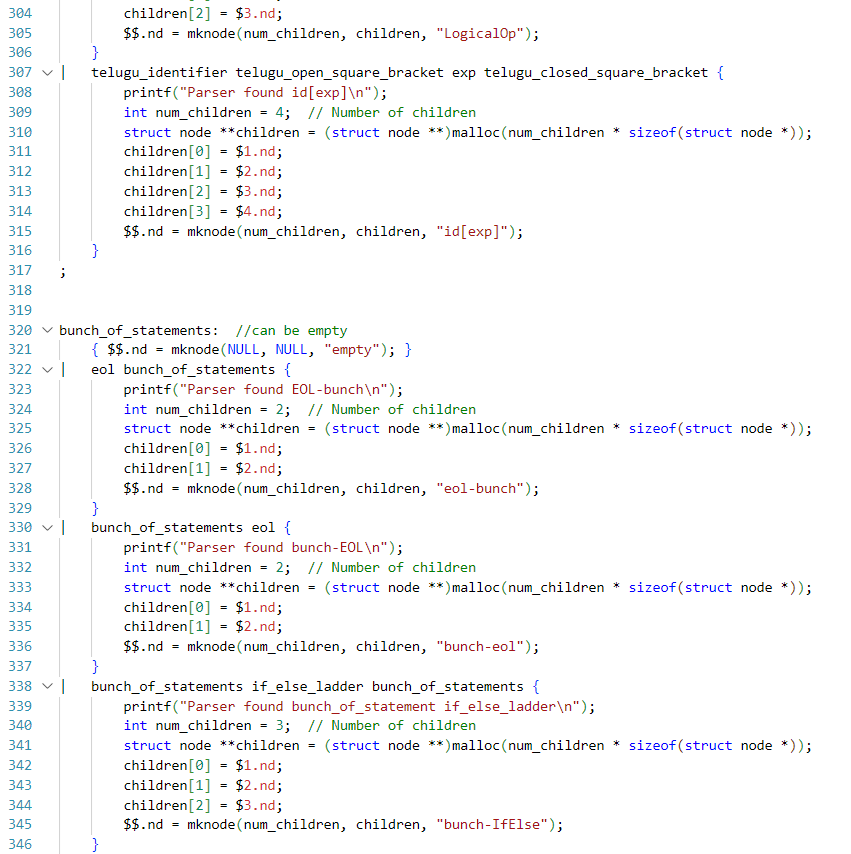
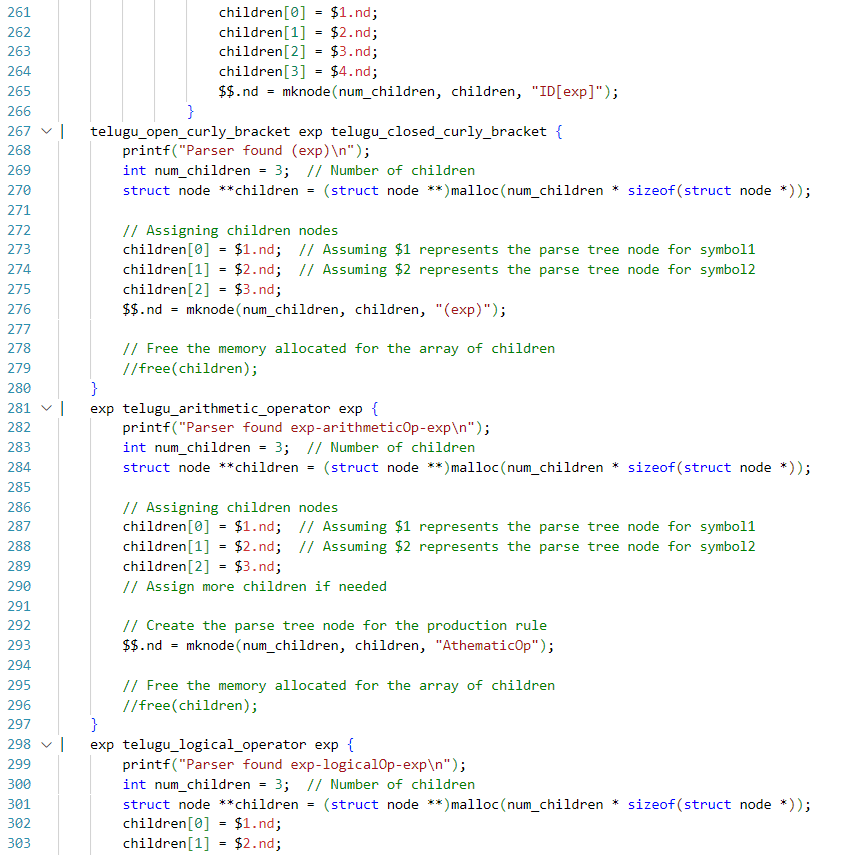
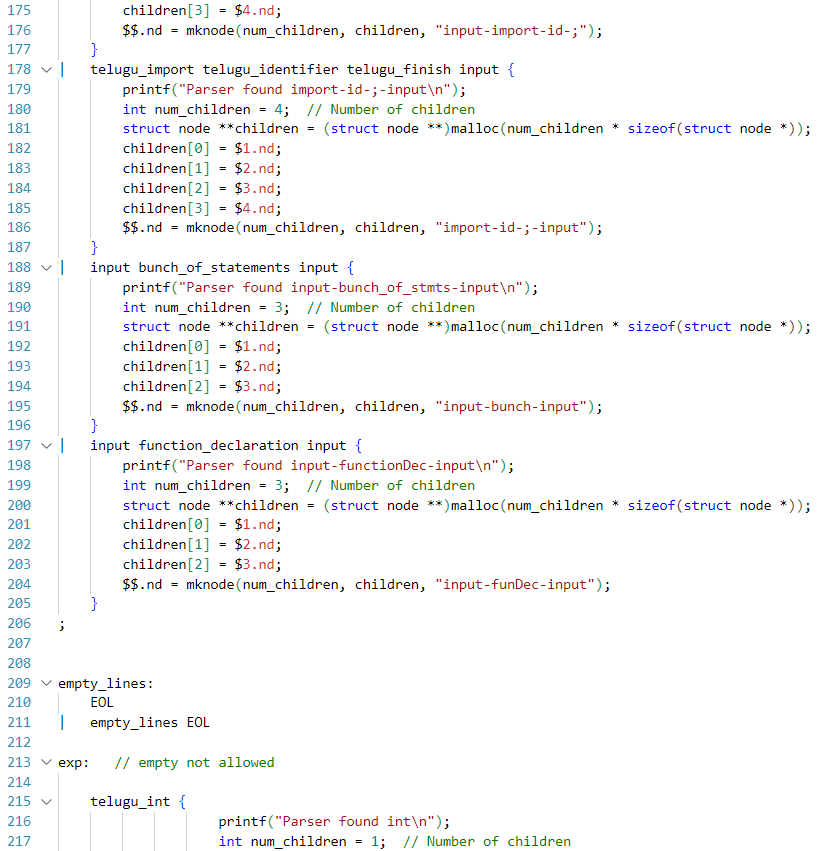
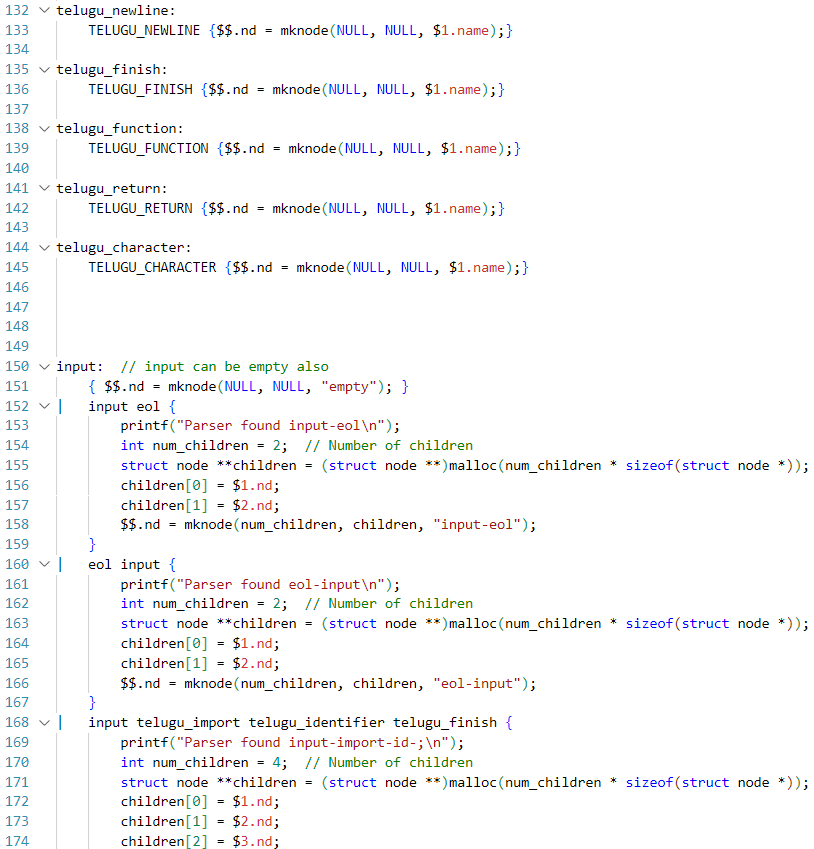
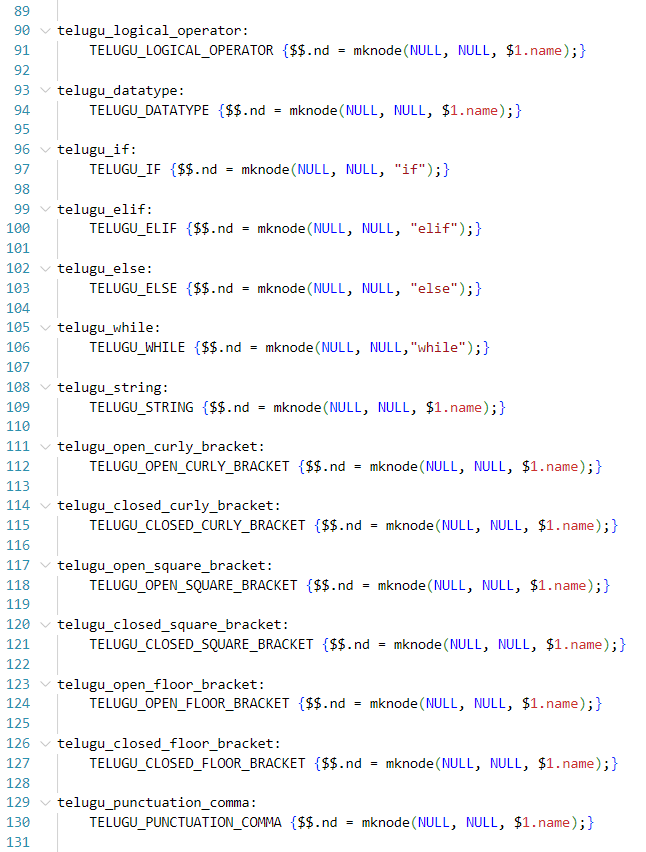
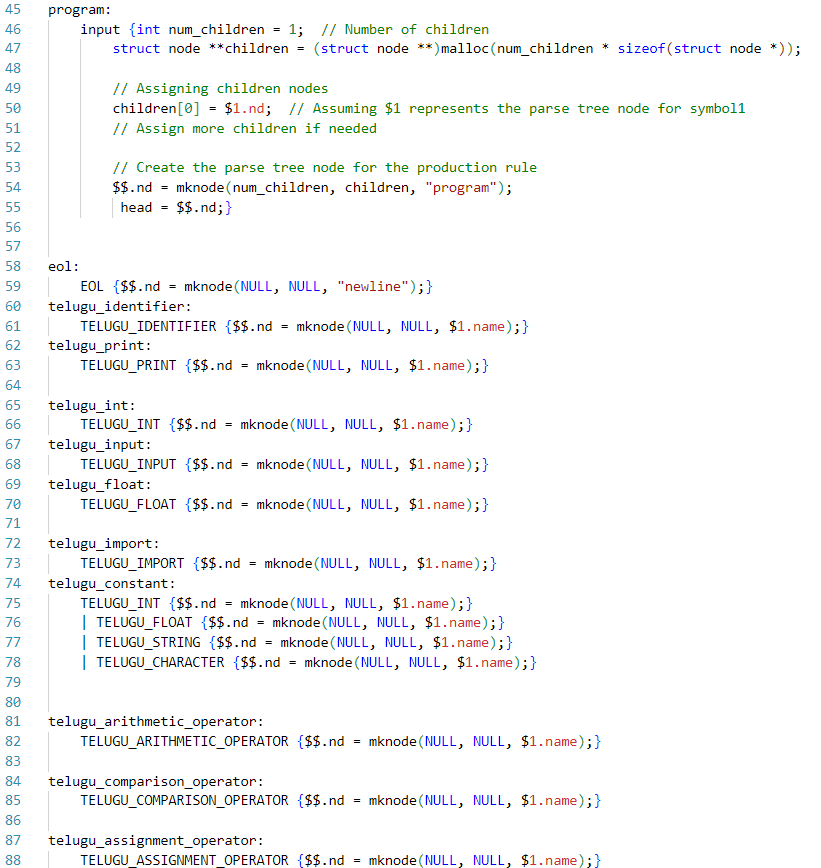
**UPDATED LEXER CODE**:





**PARSER CODE:**

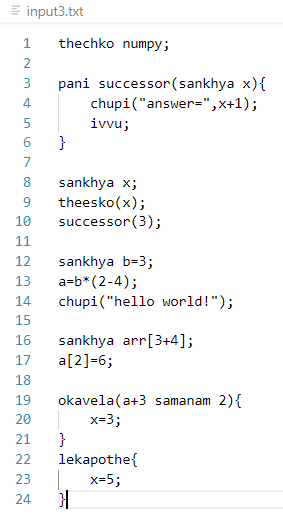




Run Commands:



**INPUT:**



// import library

// function declaration

// variable declaration and input

//equation

//print statement

// array declaration

//array indices access

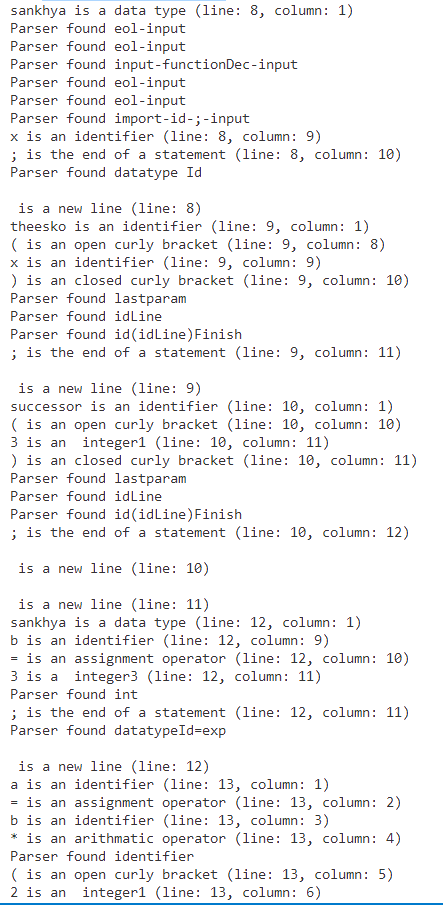
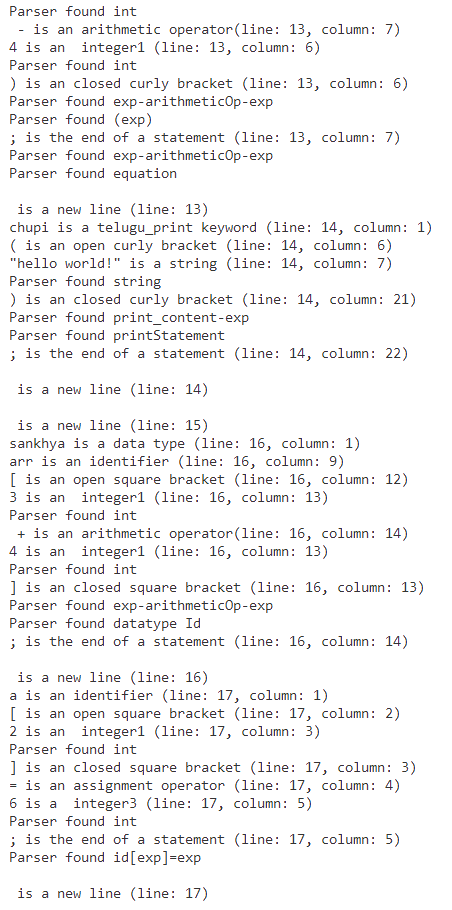
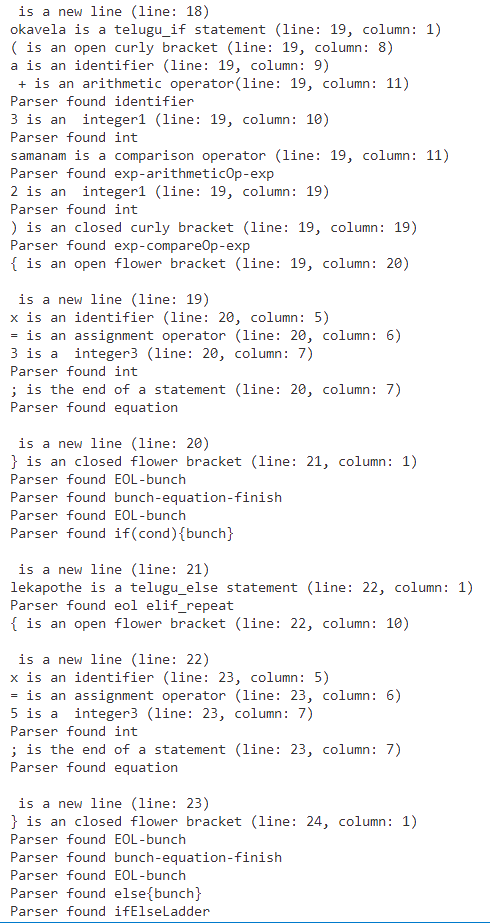
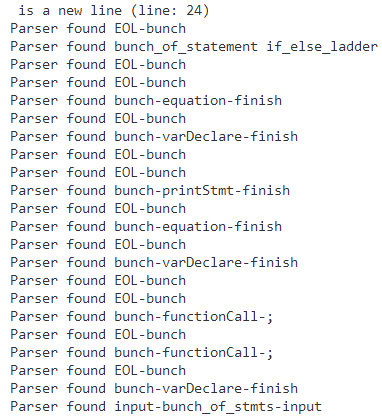
// if statement

// else statement

**OUTPUT:**

// This is Parser Output combined with Lexer Output

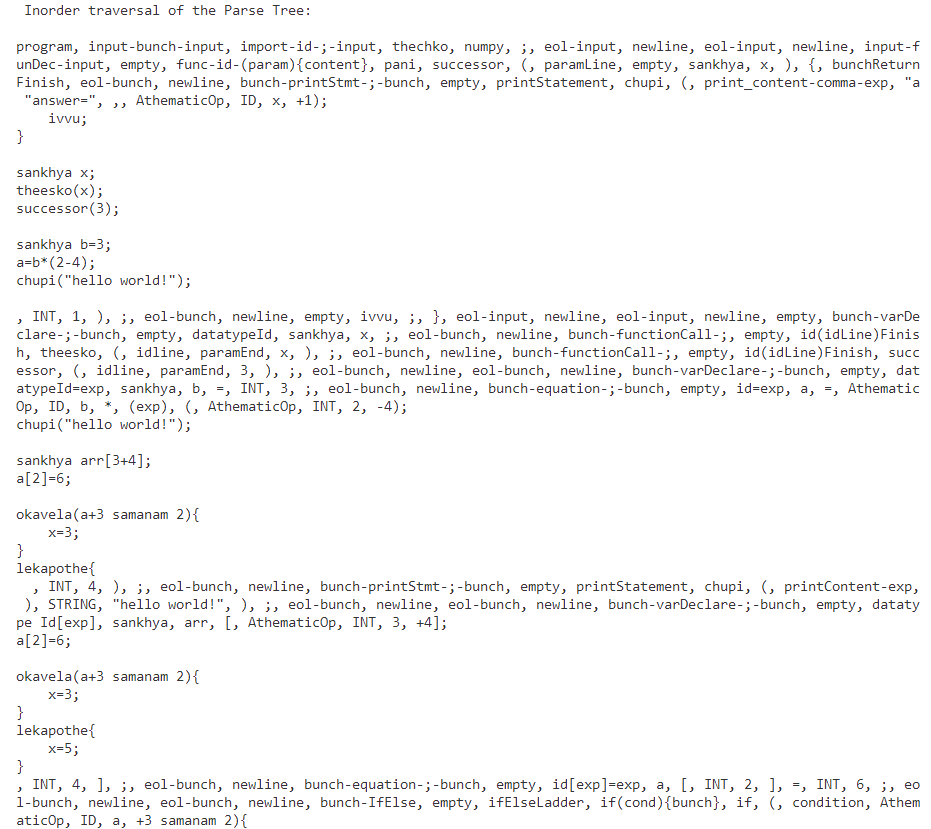
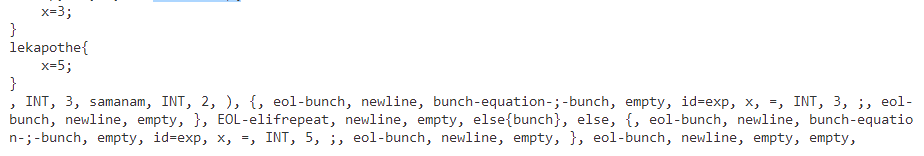


**PARSE TREE GENERATION:**

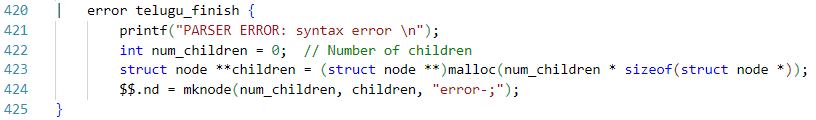
The parse tree is stored as a node with pointers to its its child nodes. We have implemented with Multiple Children to allow easier grammar incorporation.

Finally the parse tree’s traversal is printed in the order Root->ChildrenFromLeftToRight.

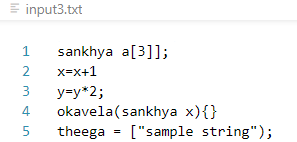
 

**ERROR HANDLING:**

The code is able to detect multiple lines of error because of

**** ****

**Input with syntax errors:**



**Detected Errors**:

