**COMPILER DESIGN LAB-1**

**TELUGU LEXICAL ANALYZER**

Team members:

**Karthik Chittoor – 106121033**

**Vishnu Vardhan P – 106121089**

**Pranav Prakash - 106121093**

**A.Components of our Programming Language(Telugu).**

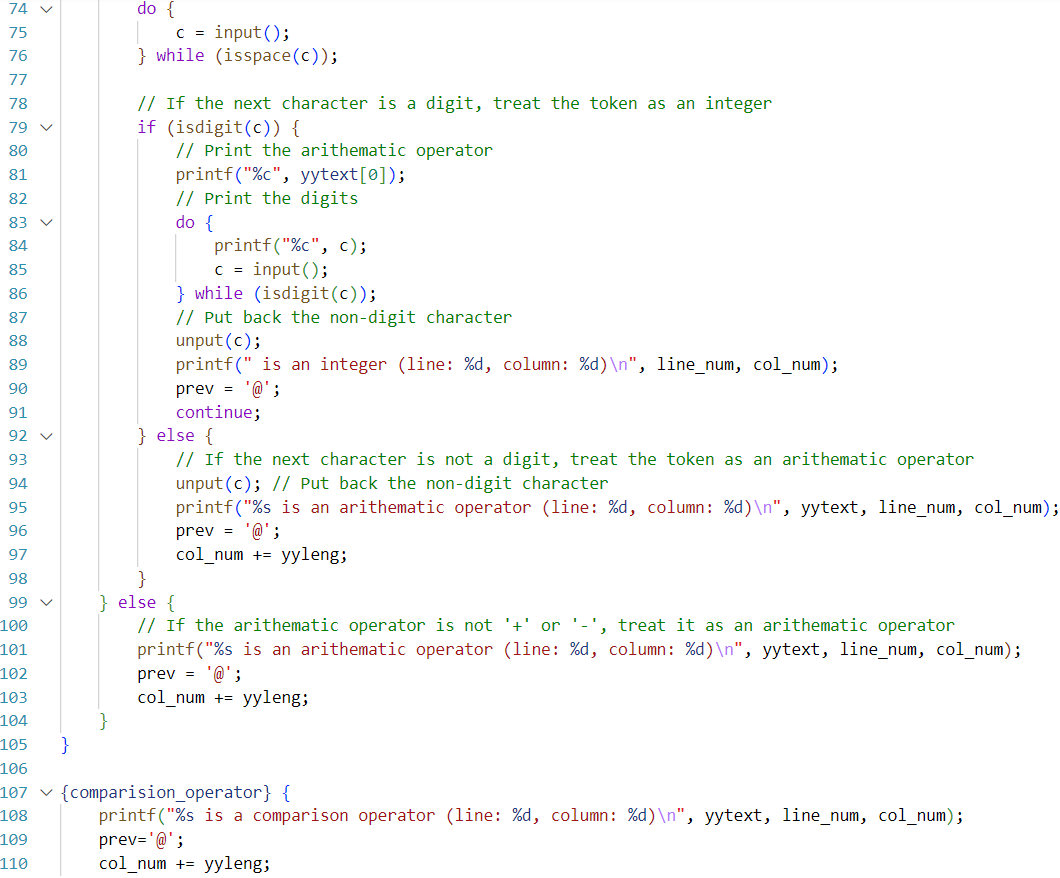
# 

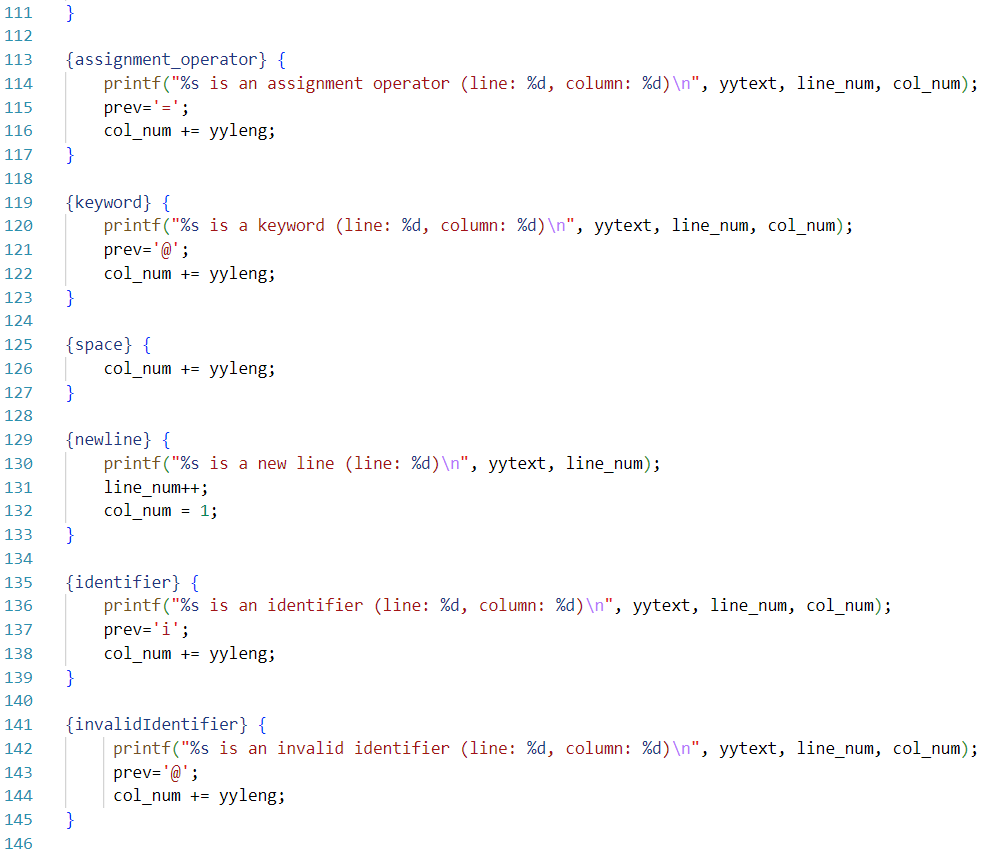
# B.Regular Expression:

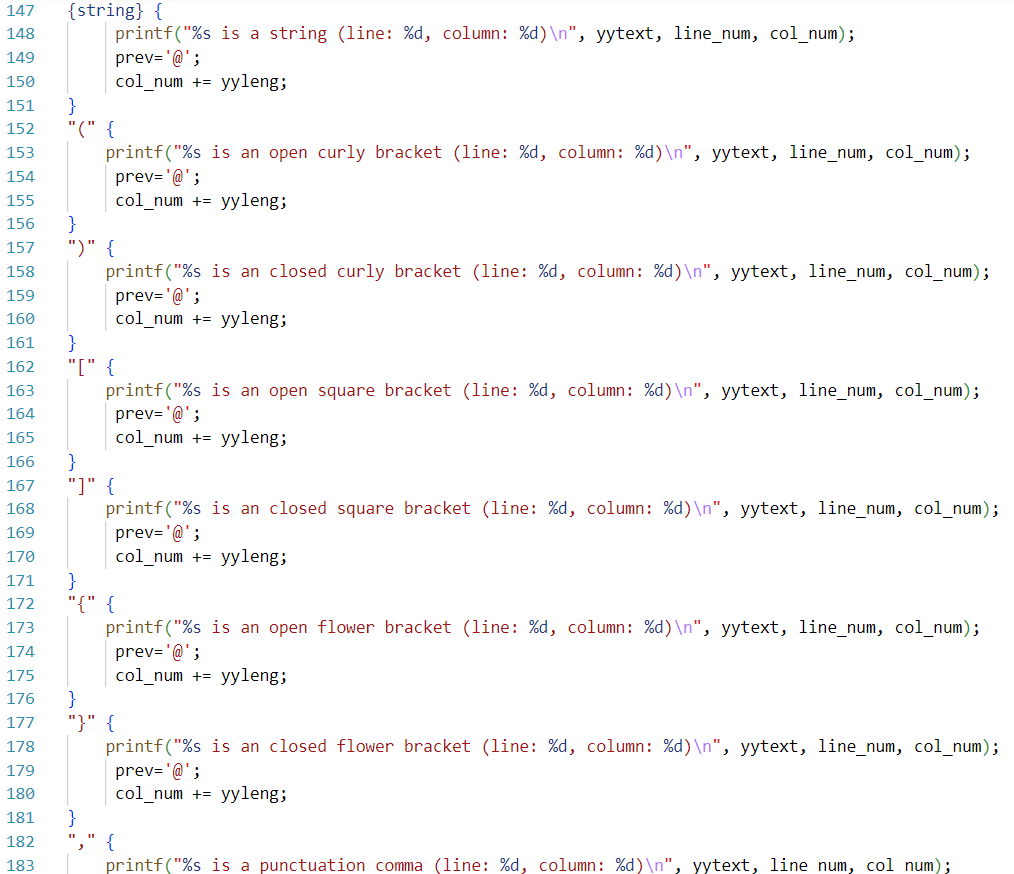
# 

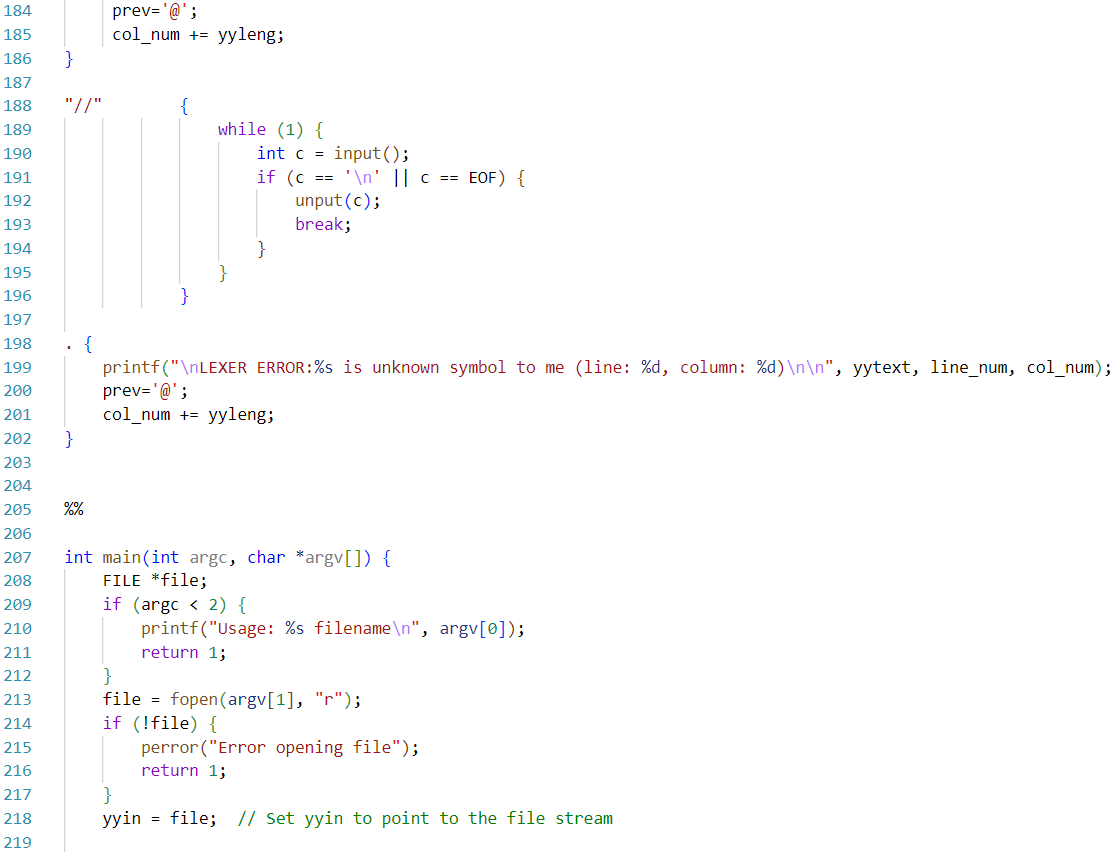
# C. LEX Code :

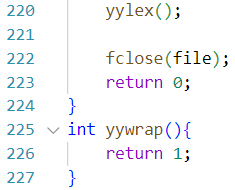
# 



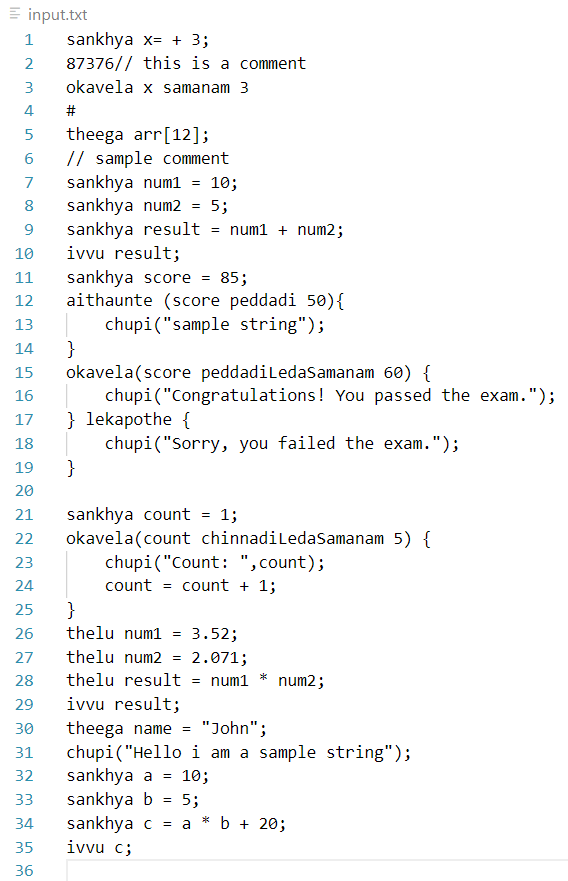




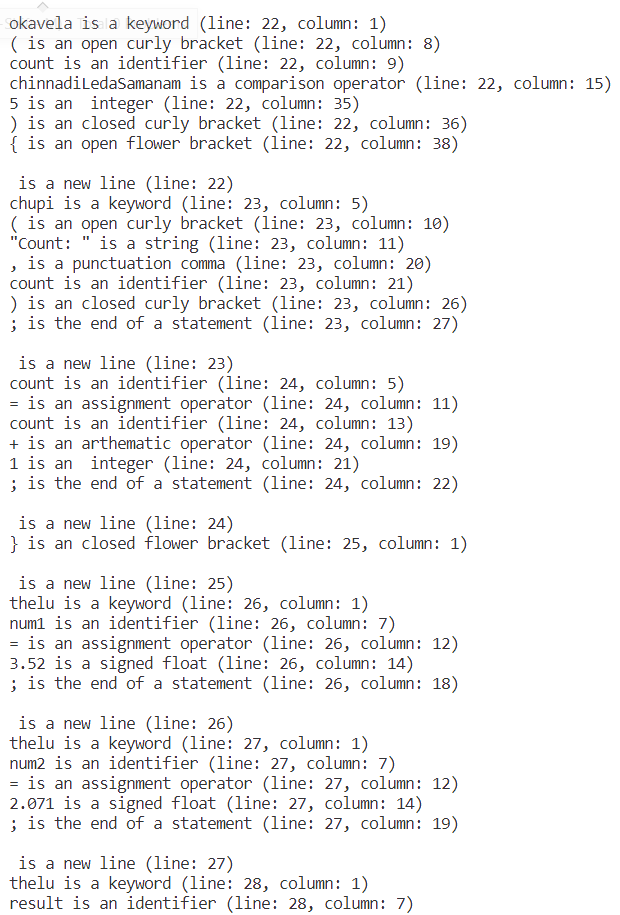
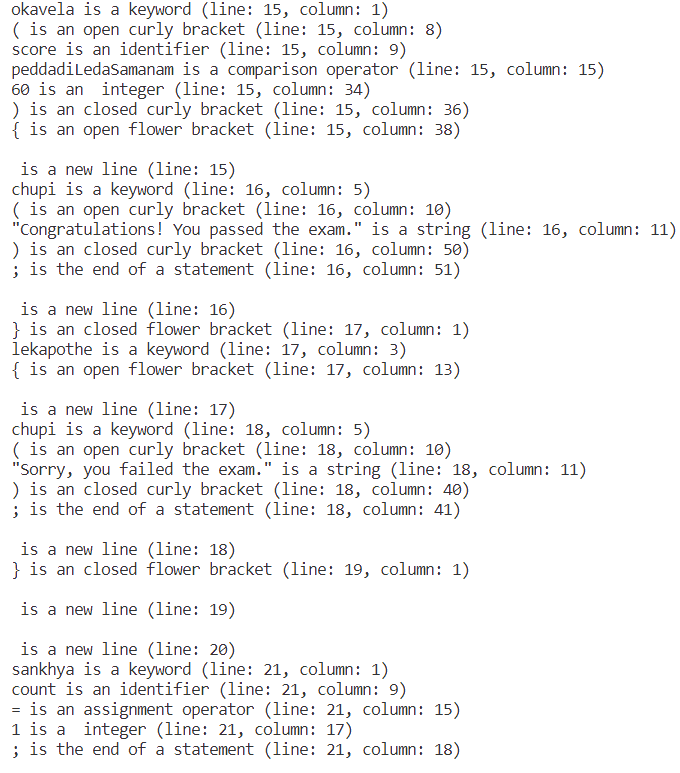
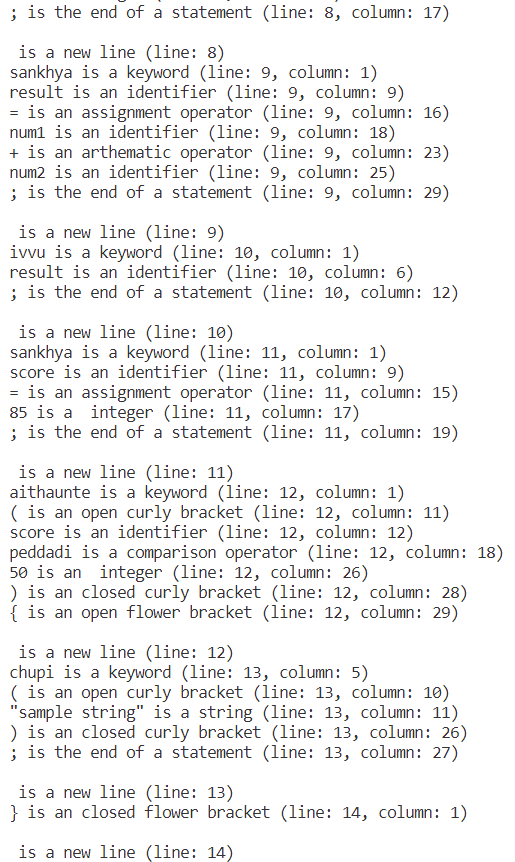
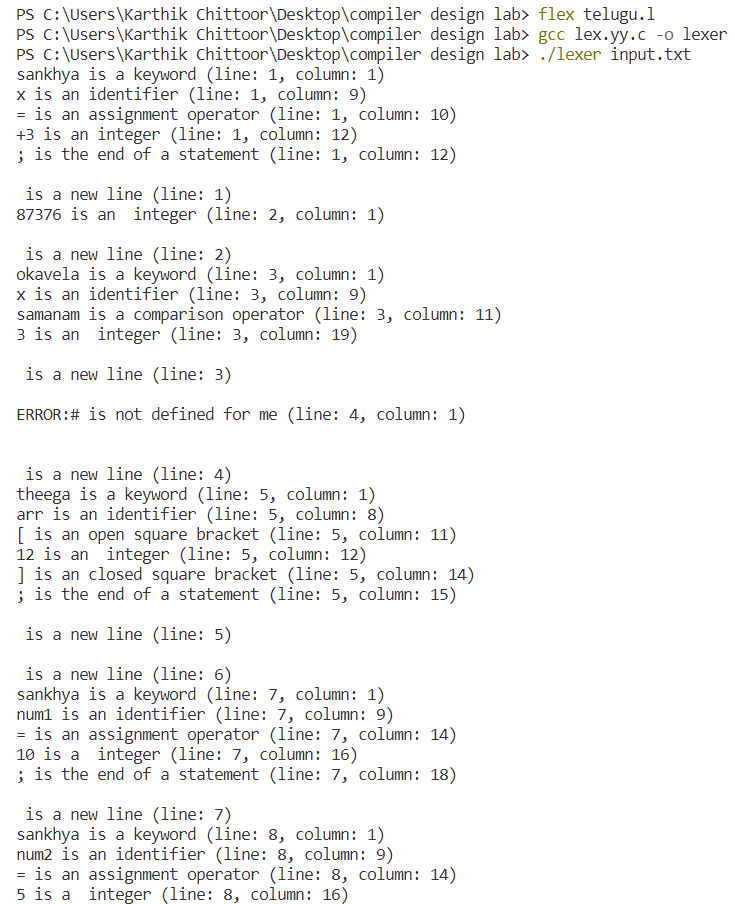


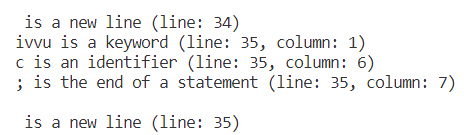
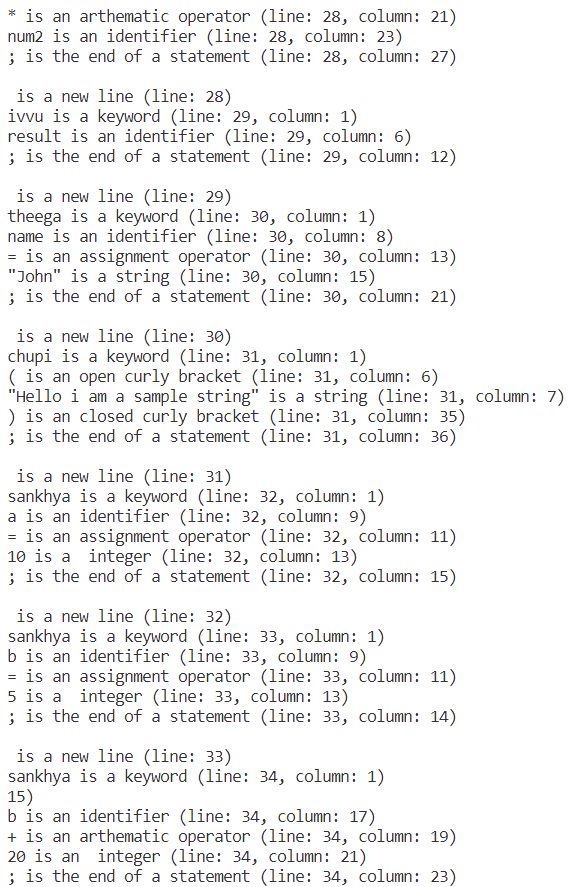


**INPUT:**

****

**OUTPUT:**

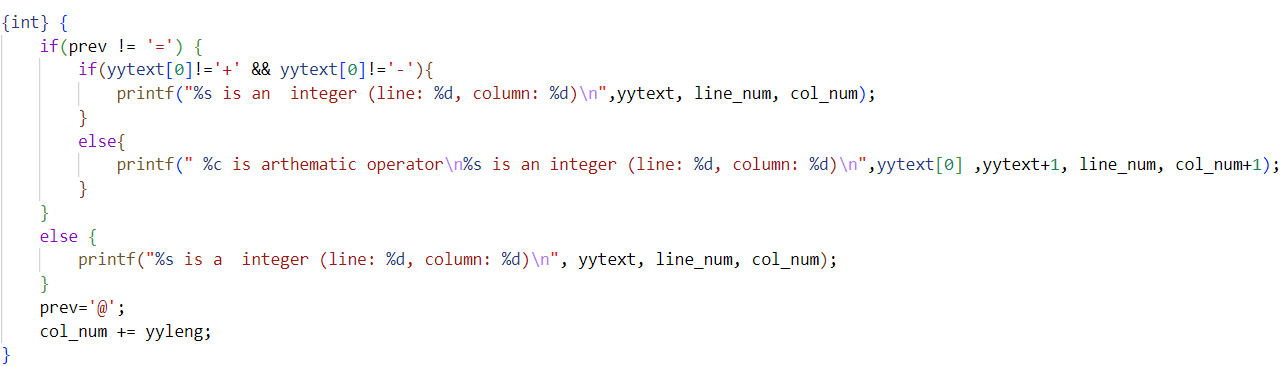


****

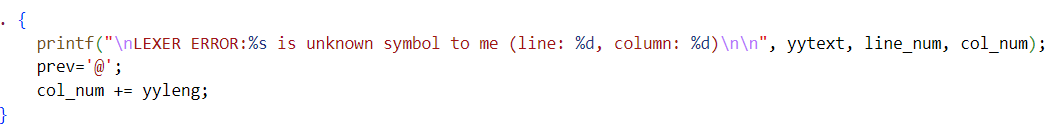
**D.Error/Ambiguity Handling:**

sankhya a=-3; // here -3 should be read as a signed integer

a=a-3; // here ‘-‘ should be read as binary arthematic operator and 3 as unsigned integer seperately

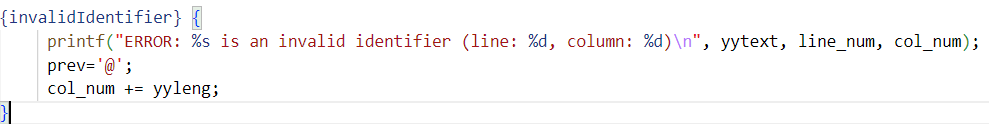
****

Here “prev” stores ‘=’ if it lexer saw an assignment operator. Other wise it will store a dummy symbol ‘@’. Based on prev as a **look-behind,** we differentiate if a + or – is unary/binary operator at that statement.



Any unknown symbol(not defined) the lexer comes across is shown as a lexer error.

****

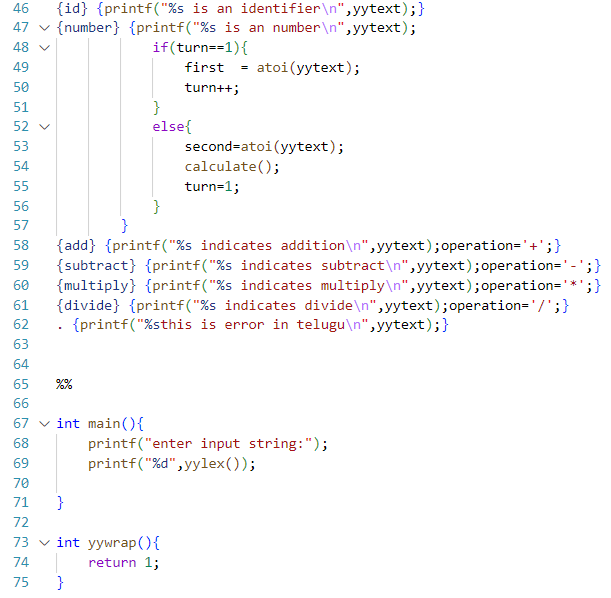
****

Identifiers that start with digits are shown as invalid Identifier error.

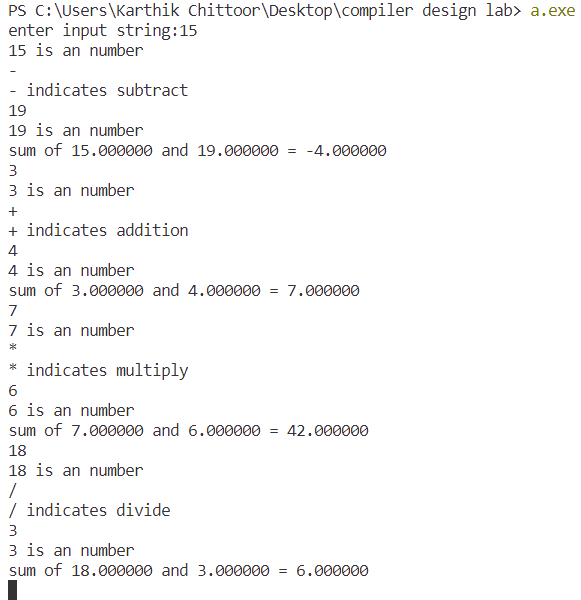
(note that we could simple shown it as error instead of categorizing it as a separate token, but this is for more information to the coder.)

**F. Small Calculator program in Telugu**

**Code:**

**** ****

**Ouput:**

****

**Result:**

The Lexical Analyser is able to generate tokens correctly based on the regular expressions defined.

1.One line comments are skipped

2.Unknown symbols are shown as error.

3.Identifiers starting with a number are shown as invalid identifiers-error().

**Conclusion:**

The lexical analyser (the first stage of compilation) has been successfully implemented in LEX.