**Lab 01**

Link to the lab 01 - <https://github.com/BuruguEric/100036_CCProjects>

**Lab 02**

A lexical analyzer or scanner is the first phase of a compiler. It is involved in the reading of the input of characters and producing a sequence of tokens for the syntax analyzer.

The following was the output of the lexical analyzer I built:

**INPUT:**

Void main ()

{

15 \* 17 = 133

}

**OUTPUT:**

The Outputs generated are basically the tokens generated from the input provided. The tokens are generated using a regex that is designed to identify mathematical expressions. The regex can therefore recognize numbers and operators. From the input, we know that *15, 17* and *133* are Numbers and *+* and *=* are operators. “**void**” is a word and “**keyword**” is a role, so both make a token which is void is a keyword.

The regular expressions used are:

1. *-* **Used to recognize Numbers**
2. *[\ –* **Used to recognize the operators**
3. *–* **Validates that the input only contains numbers and operators, otherwise it produces an error.**
4. *-* **Validate that there are no operators at the beginning of the input.**
5. *-* **Validate that there are no operators at the end of the input.**