# Lab Assignment 2- Lexical Analysis

In this assignment, you will work on implementing a lexical analyzer in your preferred programming language. We will consider some basic constructs (e.g., if-else statements), understand what are the different tokens (and patterns for the tokens) in the considered language/constructs.

Implement a lexical analyzer (using any programming language) for the considered tokens/patterns. Your program should take a statement as input and return the sequence of tokens as output.

---

Let us consider the example grammar for branching statements given below:

$$
\begin{array}{lcl}
stmt & \rightarrow & \textbf{if } expr \textbf{ then } stmt \\
 & | & \textbf{if } expr \textbf{ then } stmt \textbf{ else } stmt \\
 & | & \epsilon \\
expr & \rightarrow & term \textbf{ relop } term \\
 & | & term \\
term & \rightarrow & \textbf{id} \\
 & | & \textbf{number}
\end{array}
$$

**The patterns for the tokens in the language are described below**:

$$
\begin{array}{rcl}
digit & \rightarrow & \texttt{[0-9]} \\
digits & \rightarrow & digit^{+} \\
number & \rightarrow & digits\ (\texttt{.}\ digits)?\ (\ \texttt{E}\ \texttt{[+-]}?\ digits\ )? \\
letter & \rightarrow & \texttt{[A-Za-z]} \\
id & \rightarrow & letter\ (\ letter \mid digit\ )^{*} \\
if & \rightarrow & \texttt{if} \\
then & \rightarrow & \texttt{then} \\
else & \rightarrow & \texttt{else} \\
relop & \rightarrow & \texttt{<} \mid \texttt{>} \mid \texttt{<=} \mid \texttt{>=} \mid \texttt{=} \mid \texttt{<>}
\end{array}
$$

Implement a lexical analyzer using any programming language. Your program should take a statement as input and return the sequence of tokens as output.

---

- **Example Input**: `if input<10 then output1=100 else output2>=100`
- **Expected output**:

```
(if,)
(id,input)
(relop,<)
(number,10)
(then,)
(id,output1)
(relop,=)
(number,100)
(else,)
(id,output2)
(relop,>=)
(number,100)
```

---