Week 02

Lexical Analysis

Question 1:

Use ANTLR to write regular expression describing a Pascal **identifier** that must begin with a lowercase letter ('a' to 'z'), but may continue with many characters which are lowercase letter or digit ('0' to '9').

```
ID
        : [a-z][a-z0-9]*
        ;
```

Question 2:

A *regular definition* is used to name a regular expression and then the name is used in another regular expression. For example, given the following regular definition:
letter `[a-z]`
manyletter `letter+`
In ANTLR, to define a *regular definition*, we use fragment as the following example:

```
fragment Letter: [a-z];
Manyletter: Letter+;
```

Use *fragment* in ANTLR to rewrite the regular expression for the above token Identifier

```
fragment Letter
        : [a-z]
        ;
fragment Digit
        : [0-9]
        ;
ID
```

```
        : Letter (Letter | Digit)*
        ;
```

Question 3:

Use ANTLR to write regular expressions describing the following Pascal tokens:

- For a number to be taken as "**real**" (of "floating point") format, it must either have a decimal point, or use scientific notation.

```
fragment Digit
        : [0-9]
        ;
fragment ScientificNotation
        : [eE] [+-]? Digit+
        ;
REAL
        : Digit* '.' Digit+ ScientificNotation?
        | Digit+ '.' Digit* ScientificNotation?
        | Digit+ ScientificNotation
        ;
```

- **Strings** are made up of a sequence of characters between single quote: 'string'. The single quote itself can appear as two single quotes back to back in a string: 'isn”t'.

```
STRING
        : '\'' ( ~('\''|'\\') | '\'\'')* '\''
        ;
```

Question 4:

Find regular expressions for each of the following description:

- $\{a^n b^m | n \geq 0, m > 2\}$

```
a*bbb+
```

- $\{a^n b^m | n + m \text{ is even}\}$

```
(aa | ab | bb)+
```

- $\{a^n b | n \bmod 3 = 0\}$

```
a(aaa)*b
```