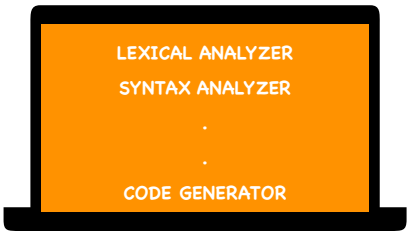
PROJECT REPORT

COMPILER DESIGN

BY MOHAMMAD SAMI &
MAHAN AHMADVAND
ID : 9531095 &
9731071
DR.RAZZAZI

PROJECT REPORT

COMPILER DESIGN



BY MOHAMMAD SAMI &
MAHAN AHMADVAND
ID : 9531095 &
9731071
DR. RAZZAZI

DR.RAZZAZI

TEACHING ASSISATNT :

MOHAMMAD ROBATI

AMIRREZA SHIRMAST

ROUZBEH GHASEMI

SEPEHR ASGARIAN

AMIRALI SAJADI

PARSA FARINNIA

FIRST PHASE

LEXICAL ANALYZER DESIGN

Lexical analyzer :

Lexical Analysis is the first phase of the compiler also known as a scanner. It converts the High level input program into a sequence of **Tokens**.

What is a token?

A lexical token is a sequence of characters that can be treated as a unit in the grammar of the programming languages.

Lexeme: The sequence of characters matched by a pattern to form the corresponding token or a sequence of input characters that comprises a single token is called a **Lexeme**.

Tokens used in the project :

'ID'

'INTEGERNUMBER'

'FLOATNUMBER'

'INTEGER', 'FLOAT'

'BOOLEAN'

'FUNCTION'

'TRUE'

'FALSE'

‘PRINT’

‘RETURN'

‘MAIN’

‘IF’

‘ELSE’

‘ELSEIF’

‘WHILE’

‘ON’

‘WHERE’

‘FOR'

‘AND’

‘OR’

‘NOT’

‘IN’

‘ASSIGN’

‘SUM’

‘SUB’

‘MUL’

‘DIV'

‘MOD’

‘GT’

‘GE’

‘LT’

‘LE’

‘EQ’

‘NE’

‘LCB’

‘RCB'

‘LRB’

‘RRB’

‘LSB’

‘RSB’

‘SEMICOLON’

‘COLON’

‘COMMA’

'ERROR'

Code Analysis :

Some Challenging Regular Expressions => :))

ID => [_a-z][a-zA-Z_0-9]*

FLOATNUMBER => [0-9]{1,9}\.[0-9]+

INTEGERNUMBER => [0-9]{1,9}

```
  ERROR => ([0-9]{10,})(\.[0-9]+)
  |([0-9]{10,})
  |([0-9]+)(\.[0-9]+){2,}
   |([A-Z])+[a-zA-Z_0-9]+
   |[0-9]+[a-z_A-Z][a-zA-
Z_0-9]+
   |[\+\-\%\/\*](\s*[\+\-\%\/
\*]+)+
   |ERROR
```

TO
BE
CONTINUED...