**Lexical analyzer Report**

****

**studentID : 20166207**

**Team member : Wongi,Baek**

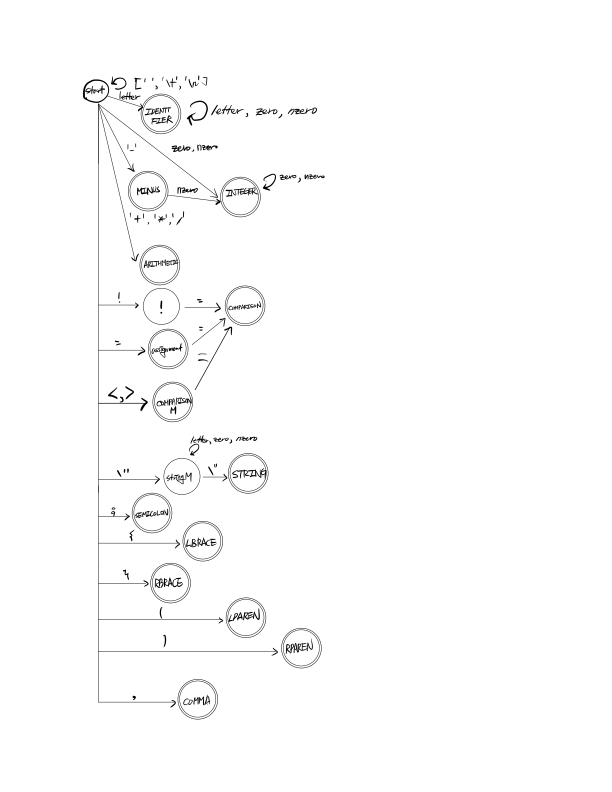
**Jungmin Choi**

**Subject : Compiler**

1. **The definition of tokens and their regular expressions**

* **Keyword = [ INT | CHAR | IF | ELSE | WHILE | RETURN ]**
* **LETTER = [ [A-Z] | [a-z] ]**
* **SPACE = [ , \n, \t]**
* **ZERO = [ 0 ]**
* **NZERO = [ 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ]**
* **ARITHMETIC = [ + | - | \* | / ]**
* **SEMICOLON = [ ; ]**
* **LBRACE = [ { ]**
* **RBRACE = [ } ]**
* **LPAREN = [ ( ]**
* **RPAREN = [ ) ]**
* **COMMA = [ , ]**
* **STRING = [ “ (letter|zero|nzero)\* “ ]**
* **COMPARISON = [(!|=|>|<)=]**
* **ASSIGNMENT = [ = ]**
* **IDENTIFIER = [ letter(letter | zero | nzero)\* ]**
* **INTEGER = [(zero|nzero)(zero|nzero)\*|(-nzero)(zero|nzero)\*]**

1. **The DFA transition graph or table for recognizing the regular expressions**



1. **All about how your lexical analyzer works for recognizing tokens**
2. **First step**

**If there is a file in same folder, we read contents of the file and then return the string**

1. **Second step**

**DFA accept each character per 1 loop. Start state is initialized by ‘start’, so transition occur by alphabet symbol. At this time, transition function is defined by dictionary. If transition function accept 1 variable, this variable has two elements character variable and alphabet and then we continue to accept symbol until finding final state and the character variable of symbol is inserted in queue . If we just find final state, we should decide that token is complete or not. So, we consider next state.**

* **1. Next state is finalstate**

**We should search nextstate**

* **2. Next state is not finalstate**

**We decide that this token is complete token!**

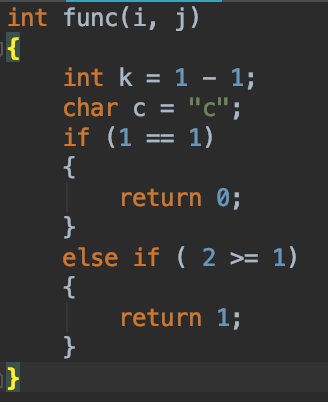
**So, Every character in queue take out and then print them.**

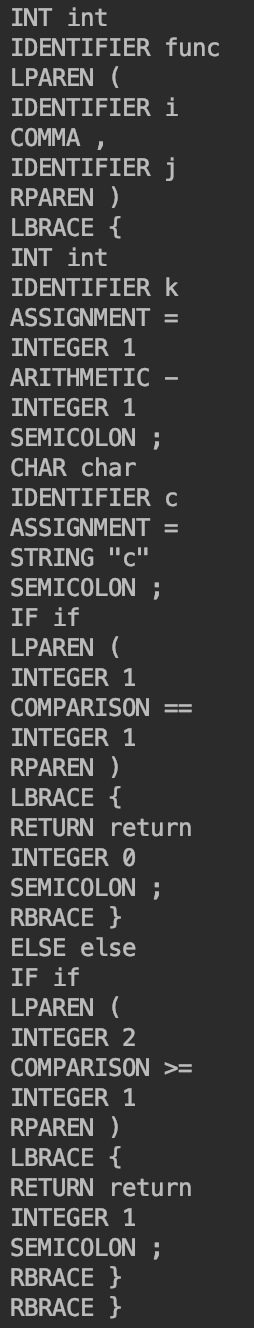
**In case of identifier, token is decided by the priority(keyword) and (+, -, \*, /) of final state is each name but their token name is ARITHMETIC.**

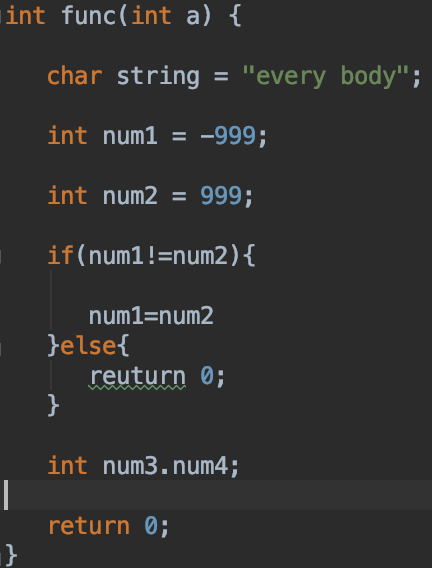
**Other case of token name is same to finalstate of name. this program also print token name**

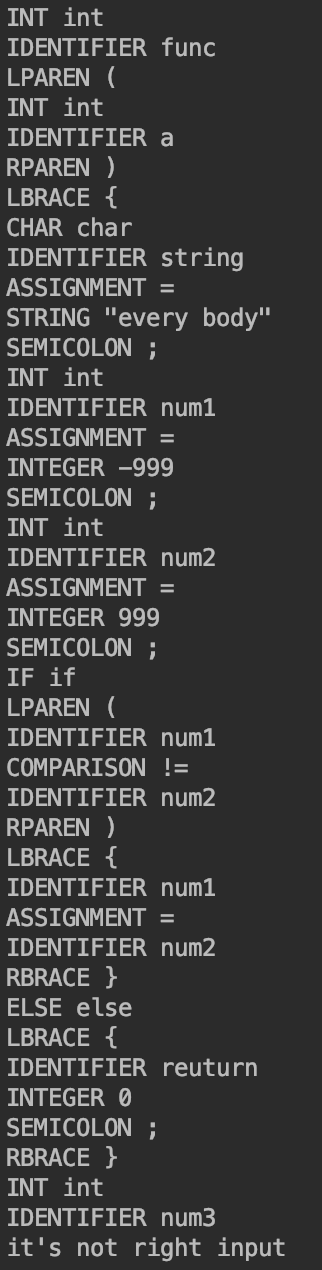
**This process repeat until meeting EOF**

**4. Practice**

****

****

****

****