**PLC – Test 2**

**Minh Dang**

1. a. Define the rules for recognizing all lexemes as their proper token, and clearly define integer token codes for each token required for this language.
2. <stmt> --> <if\_stmt> | <while\_stmt> | <as\_s> | <block>
3. <block> --> `{` { <stmt>`;` } `}`
4. <if\_stmt> -->  `if``(`<bool\_expr>`)` <stmt> [ `else` <stmt> ]
5. <while\_loop> -->  `while``(`<bool\_expr>`)` <stmt>
6. <as\_s>  --> `id` `=` <expr>
7. <factor> --> `id` | `int\_lit` | `float\_lit` | `(` <expr> `)`
9. <bool\_expr> --> <band> { `OR` <band> }
10. <band> --> <beq> { `AND` <beq> }
11. <beq> --> <brel> { (`!=`|`==`) <brel> }
12. <brel> --> <expr> { (`<=`|`>=` | `<` | `>`) <expr> }
13. <expr> --> <term> { (`+`|`-`) <term> }
14. <term> --> <not> { (`\*`|`\`|`%`) <bnot> }
15. <not> -> [!]<bfactor>
16. <factor> --> `id` | `int\_lit` | `float\_lit` | `bool\_lit` | `(` <bexpr> `)`

b. Define production rules for implementing the mathematical syntax of operators and operands, loops, variable declaration, selection statements.

1. <expr> --> <term> { (`+`|`-`) <term> }
2. <term> --> <factor> { (`\*`|`\`|`%`) <factor> }

c. (10 points) Show whether every rule set in your language conforms to the standard of an LL Grammar.

* In order to conform to the standard of an LL they must not ambiguous grammar and must pass pairwise disjoiinteness test.

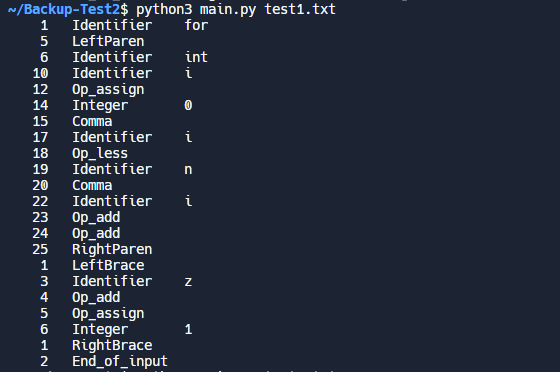
d. Implemented in the code

e. Implemented in the code

f. Implemented in the code

g. (10 points) Create 4 test files that have different names where each should have 30 or more lexemes that can be converted into tokens.

* There are 4 test file in the code name test1.txt, test2.txt, test3.txt, test4.txt
* test1.txt and test2.txt are correct.



* test3.txt is syntax error since the variable name cant start with number and word.

h. (20 points) Create a LR (1) parse table for your language. And show the trace of 4 code samples. Each must have 6 or more tokens.

1. S' -> E

E -> E + T

E -> E - T

E -> T

T -> T \* F

T -> T / F

T -> T % F

T -> F

F -> id

F -> int

F -> float

F -> (E)

* This set pass the test

