<expression> --> <term> <expression1>

<term> --> <factor> <term1>

<term1> --> <mulop> <factor> <term1>|

<mulop> --> **star\_sign** | **forward\_slash**

<factor> --> **ID** <factor1> | **NUMBER | minus\_sign NUMBER** | **left\_parenthesis** <expression>**right\_parenthesis**

<factor1> -->|**left\_bracket**<expression> **right\_bracket|left\_parenthesis** <expr list> **right\_parenthesis**

<expression1> --> <addop> <term><expression1>|

<addop> --> **plus\_sign** | **minus\_sign**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<block statements> --> **left\_brace**<statements> **right\_brace**

<statements> --> **empty** | <statement> <statements>

<statement> --> <assignment>| <func call>| <if statement> | <while statement> | <return statement> | <break statement> | <continue statement> | **read left\_parenthesis  ID right\_parenthesis semicolon**| **write left\_parenthesis**<expression> **right\_parenthesis semicolon** | **print left\_parenthesis  STRING right\_parenthesis semicolon**

<assignment> --> <id> **equal\_sign** <expression> **semicolon**

<func call> --> **ID left\_parenthesis**<expr list> **right\_parenthesis semicolon**

<expr list> --> **empty** | <non-empty expr list>

<non-empty expr list> --> <expression><non-empty expr list1>

<non-empty expr list1> -->**comma** <expression><non-empty expr list1>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<if statement> --> **if** **left\_parenthesis**<condition expression>**right\_parenthesis**<block statements>

<condition expression> --> <condition><condition expression1>

<condition expression1> --> **empty** |  <condition op> <condition>

<condition op> --> **double\_and\_sign** | **double\_or\_sign**

<condition> --> <expression> <comparison op> <expression>

<comparison op> --> **==** | **!=** | **>** | **>= | < | <=**

<while statement> --> **while left\_parenthesis**<condition expression> **right\_parenthesis** <block statements>

<return statement> --> **return**<return statement1>

<return statement1> --> <expression> **semicolon |semicolon**

<break statement> ---> **break semicolon**

**<**continue statement> ---> **continue semicolon**

<program> --> <data decls> <func list>

<func list> --> **empty**| <func> <func list>

<func> --> <func decl><func1>

<func1> -->**semicolon** | **left\_brace**<data decls> <statements> **right\_brace**

<func decl> --> <type name> **ID** **left\_parenthesis**<parameter list> **right\_parenthesis**

<type name> --> **int** | **void | binary | decimal**

<parameter list> --> **empty** | **void** | <non-empty list>

<non-empty list> --> <type name> **ID** <non-empty list1>

<non-empty list1> --> **comma** <type name> **ID** <non-empty list1>| **empty**

<data decls> --> **empty** | <type name> <id list> **semicolon** <data decls>

<id list> --> <id><id list1>

<id list1> -->**comma**<id><id list1>| **empty**

<id> --> **ID**<id1>

<id1> -->  **empty** |**left\_bracket** <expression>**right\_bracket**