**Code Generation Assignment on Ven**

Name: kareem gamal Mahmoud Mohamed

Computer science level 4

Advanced compiler

The assignment contains :

-Scanner

-Parser

- EBNF syntax

-Semantic Analyzer

-Code Generation

-Test File

-Abstract Syntax

**EBNF syntax for ven language :**

<parse\_ven> ::= <program> EOF

<program> ::= <block>

<block> ::= begin <declaration-seq> <command-seq> end<declaration-seq> ::= <declaration> {<declaration>}<declaration> ::= <type> <name-list> | proc <name> [(<parameter-list>)] = <command>

<type> ::= integer | Boolean<parameter-list> ::= <type> <name-list> { ; <type> <name-list>}

<name-list> ::= <name> { , <name>}<command-seq> ::= <command> { ; <command>}<command> ::= <name> :=

<expr> | read <name> | write <expr> | if <expr> then

<command-seq> [else <command-seq>] end if |

while <expr> do <command-seq> end while | call <name> [(<name-list>)]

<expr> ::= <expr1> {or <expr1>}<expr1> ::= <expr2> {and <expr2>}

<expr2> ::= <expr3> | not <expr>

<expr3> ::= <expr4> {<relation> <expr4>}<expr4> ::= <term> {<weak op> term}<term> ::= <element> {<strong op> <element>}

<element> ::= [-] <elem>

<elem> ::= <numeral> | <name> | (<expr>)

<relation> ::= < | <= | <> | = | > | >=

<weak op> ::= + | –

<strong op> ::= \* | /<name> ::= [<ident>] <letter> | <ident> <digit>

<letter> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

<numeral> ::= <digit> {<digit>}

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

**Abstract Syntax for ven:**

Program 🡨 Program(Block)

Block 🡨 Block(DecSeq , ComSeq)DecSeq 🡨 Declaration+ SB e

Declaration 🡨 var(type , Ids) | proc( Id , Paras? , command)type 🡨 integer | BooleanParas 🡨 Paras (type , Ids)+ SB ;

Ids 🡨 Id+ SB ,

ComSeq 🡨 command+ SB ;Command 🡨 Assign(Id , Expr) | Read(Id) | Write(Id) |

IfThenElse (Expr , ComSeq , ComSeq?)

| While(Expr , ComSeq) | Call(Id , Ids?)

Expr 🡨 Expr1 + SB or

Expr1 🡨 Expr2 + SB and

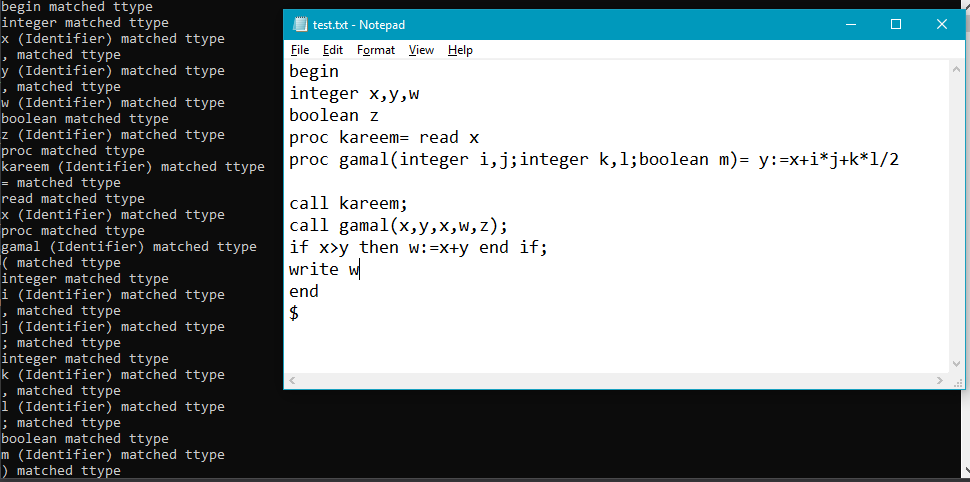
Expr2 🡨 Expr3 | Not(Expr)

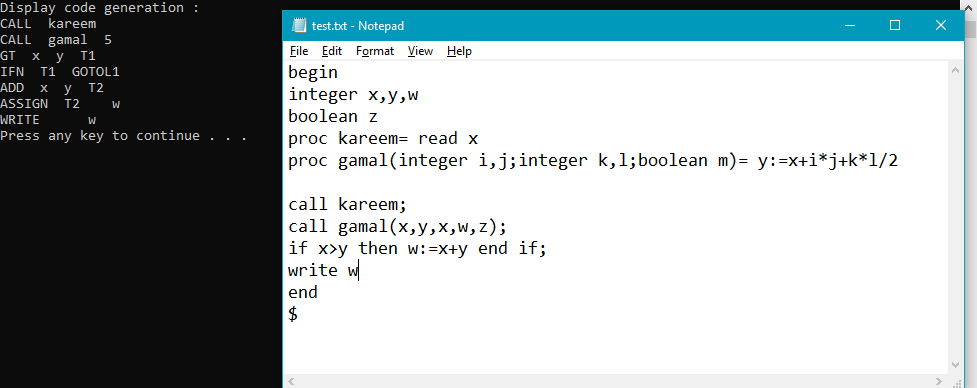
Expr3 🡨 Expr4 + SB relation

Expr4 🡨 Term + SB w\_op

Element 🡨 Number | Id | Expr | -(Element)

<relation> ::= < | <= | <> | = | > | >= <weak op> ::= + | – <strong op> ::= \* | /

**Run the test file :**

 This is a part of parser and check semantic with two pases.

This is the code generation for this program.