Ubuntu

```
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs370/lab6$ make
yacc -d lab6.y
lab6.y: warning: 1 shift/reduce conflict [-Wconflicts-sr]
lex lab6.l
gcc lex.yy.c y.tab.c ast.c -o lab6
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs370/lab6$ ls
Makefile  ast.c  ast.h  fulltest  lab6  lab6.l  lab6.y  lab6test.al  lex.yy.c  run.sh  test  y.tab.c  y.tab.h
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs370/lab6$ clear
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs370/lab6$ make
yacc -d lab6.y
lab6.y: warning: 1 shift/reduce conflict [-Wconflicts-sr]
lex lab6.l
gcc lex.yy.c y.tab.c ast.c -o lab6
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs370/lab6$ ./lab6 < lab6test.al
Variable VOID x
Variable INT x[100]
INT FUNCTION main
 (VOID)
 BLOCK STATMENT
  Variable INT x
  BLOCK STATMENT
   Variable INT y
   WHILE Statement
    EXPR <=
     EXPR +
      IDENTIFIER x
      EXPR /
       Number with value 5
       Number with value 2
     EXPR -
      EXPR +
       Number with value 2
       IDENTIFIER z
      Number with value 5
     IF Statement
      EXPR >=
       EXPR -
        IDENTIFIER h
        Number with value 2
       EXPR -
        Number with value 3
        Number with value 2
     Then
      READ STATMENT
       IDENTIFIER x
        Array Reference [
         Number with value 100
        ] end of array
     Else
      WRITE STATMENT
       EXPR +
        IDENTIFIER x
         Array Reference [
          Number with value 100
         ] end of array
        Number with value 200
```

Ubuntu

```
   EXPRSTMT
    CALL FUNCTION f
     ARG
      EXPR +
       Number with value 3
       IDENTIFIER x
        Array Reference [
         IDENTIFIER x
          Array Reference [
           Number with value 100
          ] end of array
         ] end of array
     ARG
      EXPR +
       IDENTIFIER bar
       Number with value 200
     ARG
      Number with value 20
   Returning
   Returning
    EXPR +
     EXPR +
      IDENTIFIER x
      Number with value 5
     Number with value 7
   IF Statement
    EXPR >
     EXPR +
      IDENTIFIER x
      Number with value 10
     EXPR *
      Number with value 10
      Number with value 20
   Then
    Assignment STATMENT
     IDENTIFIER x
     EXPR !=
      IDENTIFIER x
      Number with value 10
   Else
   WRITE STATMENT
    Not
     EXPR OR
      EXPR AND
       Number with value 3
       Number with value 5
      EXPR AND
       Number with value 1
       Not
        Number with value 0
VOID FUNCTION f
 (
 Paramater INT  x
 Paramater INT  y
 Paramater VOID  z
 )
 BLOCK STATMENT
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs370/lab6$
```