For all runs the program arguments are: -f test.txt -l -a -v

**Error 1: Use = instead of :=**

Input file:

Output:

**Error 2: = must be followed by a number**

Input file:

Output:

**Error 3: Identifier must be followed by =**

Input file:

Output:

**Error 4: const, var, procedure must be followed by identifier**

Input file:

var

begin

x := y + 56;

end.

Output:

Source Program:

var

begin

x := y + 56;

end.

Lexeme Table:

Lexeme Token Type

var 29

begin 21

x 2

:= 20

y 2

+ 4

56 3

; 18

end 22

. 19

Lexeme List:

29 21 2 x 20 2 y 4 3 56 18 22 19

var\_sym begin\_sym ident\_sym x becomes\_sym ident\_sym y plus\_sym num\_sym 56 semicol\_sym end\_sym period\_sym

ERROR: const, var, procedure must be followed by identifier.

Parser has encountered an error...

**Error 5: Semicolon or comma missing**

Input file:

var x, y

begin

x := y + 56;

end.

Output:

Source Program:

var x, y

begin

x := y + 56;

end.

Lexeme Table:

Lexeme Token Type

var 29

x 2

, 17

y 2

begin 21

x 2

:= 20

y 2

+ 4

56 3

; 18

end 22

. 19

Lexeme List:

29 2 x 17 2 y 21 2 x 20 2 y 4 3 56 18 22 19

var\_sym ident\_sym x comma\_sym ident\_sym y begin\_sym ident\_sym x becomes\_sym ident\_sym y plus\_sym num\_sym 56 semicol\_sym end\_sym period\_sym

ERROR: Semicolon or comma missing.

Parser has encountered an error...

**Error 6: Incorrect symbol after procedure declaration**

Input file:

Output:

**Error 7: Statement expected**

Input file:

Output:

**Error 8: Incorrect symbol after statement part in block**

Input file:

var x;

begin.

end.

Output:

Source Program:

var x;

begin.

end.

Lexeme Table:

Lexeme Token Type

var 29

x 2

; 18

begin 21

. 19

end 22

. 19

Lexeme List:

29 2 x 18 21 19 22 19

var\_sym ident\_sym x semicol\_sym begin\_sym period\_sym end\_sym period\_sym

ERROR: Incorrect symbol after statement part in block.

Parser has encountered an error...

**Error 9: Period expected**

Input file:

Output:

**Error 10: Semicolon between statements missing**

Input file:

Output:

**Error 11: Undeclared identifier**

Input file:

var y;

begin

x := y + 56;

end.

Output:

Source Program:

var y;

begin

x := y + 56;

end.

Lexeme Table:

Lexeme Token Type

var 29

y 2

; 18

begin 21

x 2

:= 20

y 2

+ 4

56 3

; 18

end 22

. 19

Lexeme List:

29 2 y 18 21 2 x 20 2 y 4 3 56 18 22 19

var\_sym ident\_sym y semicol\_sym begin\_sym ident\_sym x becomes\_sym ident\_sym y plus\_sym num\_sym 56 semicol\_sym end\_sym period\_sym

ERROR: Undeclared identifier.

Parser has encountered an error...

**Error 12: Assignment to constant or procedure is not allowed**

Input file:

Output:

**Error 13: Assignment operator expected**

Input file:

Output:

**Error 14: call must be followed by an identifier**

Input file:

Output:

**Error 15: Call of a constant or variable is meaningless**

Input file:

Output:

**Error 16: then expected**

Input file:

var x, y;

begin

x:= 4;

y:= 5;

if y > x

y:= y + 1;

end.

Output:

Source Program:

var x, y;

begin

x:= 4;

y:= 5;

if y > x

y:= y + 1;

end.

Lexeme Table:

Lexeme Token Type

var 29

x 2

, 17

y 2

; 18

begin 21

x 2

:= 20

4 3

; 18

y 2

:= 20

5 3

; 18

if 23

y 2

> 13

x 2

y 2

:= 20

y 2

+ 4

1 3

; 18

end 22

. 19

Lexeme List:

29 2 x 17 2 y 18 21 2 x 20 3 4 18 2 y 20 3 5 18 23 2 y 13 2 x 2 y 20 2 y 4 3 1 18 22 19

var\_sym ident\_sym x comma\_sym ident\_sym y semicol\_sym begin\_sym ident\_sym x becomes\_sym num\_sym 4 semicol\_sym ident\_sym y becomes\_sym num\_sym 5 semicol\_sym if\_sym ident\_sym y gtr\_sym ident\_sym x ident\_sym y becomes\_sym ident\_sym y plus\_sym num\_sym 1 semicol\_sym end\_sym period\_sym

ERROR: then expected.

Parser has encountered an error...

**Error 17: Semicolon or } expected**

Input file:

Output:

**Error 18: do expected**

Input file:

Output:

**Error 19: Incorrect symbol following statement**

Input file:

Output:

**Error 20: Relational operator expected**

Input file:

Output:

**Error 21: Expression must not contain a procedure identifier**

Input file:

Output:

**Error 22: Right parenthesis missing**

Input file:

Output:

**Error 23: The preceding factor cannot begin with this symbol**

Input file:

Output:

**Error 24: An expression cannot begin with this symbol**

Input file:

Output:

**Error 25: This number is too large**

Input file:

var x;

begin

x := 999999;

end.

Output:

Error: Number [99999] at 3:7 exceeds max size...

ERROR: Unexpected end of file.

Parser has encountered an error...

**Error 26: Unexpected end of file**

Input file:

var x;

begin

x := 999999;

end.

Output:

Error: Number [99999] at 3:7 exceeds max size...

ERROR: Unexpected end of file.

Parser has encountered an error...

**Error 27: := expected**

Input file:

var x, y;

begin

x = y + 56;

end.

Output:

Source Program:

var x, y;

begin

x = y + 56;

end.

Lexeme Table:

Lexeme Token Type

var 29

x 2

, 17

y 2

; 18

begin 21

x 2

= 9

y 2

+ 4

56 3

; 18

end 22

. 19

Lexeme List:

29 2 x 17 2 y 18 21 2 x 9 2 y 4 3 56 18 22 19

var\_sym ident\_sym x comma\_sym ident\_sym y semicol\_sym begin\_sym ident\_sym x eql\_sym ident\_sym y plus\_sym num\_sym 56 semicol\_sym end\_sym period\_sym

ERROR: := expected.

Parser has encountered an error...

**Error 28: Symbol max exceeded**

Input file:

Output:

**Error 29: Code length exceeded**

Input file:

Output: