

### Test Case # 1:

```
int a print("hello") while $$
```

```
INFO Lexer - Lexing program 1...
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - I_TYPE [ int ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - PRINT_START [ print ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - StringExpr [ " ] found on line 1
DEBUG Lexer - ID [ h ] found on line 1
DEBUG Lexer - ID [ e ] found on line 1
DEBUG Lexer - ID [ l ] found on line 1
DEBUG Lexer - ID [ l ] found on line 1
DEBUG Lexer - ID [ o ] found on line 1
DEBUG Lexer - StringExpr [ " ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - WhileStatement [ while ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - EOP [ $ ] found on line 1
INFO Lexer - Lex completed with 0 errors

WARNING Lexer - Warning: line 1 - Extra "$".
Program will continue to execute assuming you meant to do this...

INFO Lexer - Lexing program 2...
DEBUG Lexer - EOP [ $ ] found on line 1
INFO Lexer - Lex completed with 0 errors
```

### Test Case # 2:

```
boolean if intP string $ * / * () {p}
```

```
INFO Lexer - Lexing program 1...
DEBUG Lexer - I_TYPE [ boolean ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - IfStatement [ if ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - I_TYPE [ int ] found on line 1
ERROR Lexer - Error: line 1 Unrecognized Token: P
Please reference grammar guide for this language.
Perhaps you tried to type an invalid character outside of the letters a through z?
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - I_TYPE [ string ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - EOP [ $ ] found on line 1
Error Lexer - Lex failed with 1 error(s)

INFO Lexer - Lexing program 2...
DEBUG Lexer - SPACE [ ] found on line 1
ERROR Lexer - Error: line 1 Unrecognized Token: *
Please reference grammar guide for this language.
Perhaps you tried to type an invalid character outside of the letters a through z?
DEBUG Lexer - SPACE [ ] found on line 1
ERROR Lexer - Error: Unterminated comment starting on line 2. Lexing terminated due to fatal error.
Please add '*' to end your comment.
Error Lexer - Lex failed with 2 error(s)
```

### Test Case # 3:

```

in a d
while
int $
boolean
string

```

```

INFO Lexer - Lexing program 1...
DEBUG Lexer - ID [ i ] found on line 1
DEBUG Lexer - ID [ n ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ d ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 2
DEBUG Lexer - WhileStatement [ while ] found on
line 2
DEBUG Lexer - SPACE [ ] found on line 2
DEBUG Lexer - I_TYPE [ int ] found on line 3
DEBUG Lexer - SPACE [ ] found on line 3
DEBUG Lexer - EOP [ $ ] found on line 3
INFO Lexer - Lex completed with 0 errors

INFO Lexer - Lexing program 2...
DEBUG Lexer - SPACE [ ] found on line 3
DEBUG Lexer - I_TYPE [ boolean ] found on line 4
DEBUG Lexer - SPACE [ ] found on line 5
DEBUG Lexer - I_TYPE [ string ] found on line 5
DEBUG Lexer - SPACE [ ] found on line 5
ERROR Lexer - Error: last line of program - Please
complete program with "$" as your last character.
Error Lexer - Lex failed with 1 error(s)

```

#### Test Case # 4:

```

{ }$
{ { { { { { { { { { { } } } } } } } } } }$
{ { { { { { { { { { { } } } } } } } } } } /* comments are ignored */ } } } }$
{ /* comments are still ignored */ int @ }$
int a
a = a
string b
a = b
} }$

```

```

INFO Lexer - Lexing program 1...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 1
DEBUG Lexer - EOP [ $ ] found on line 1
INFO Lexer - Lex completed with 0 errors

INFO Lexer - Lexing program 2...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 2
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 2
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 2
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 2
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 2
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 2
DEBUG Lexer - EOP [ $ ] found on line 2
INFO Lexer - Lex completed with 0 errors

INFO Lexer - Lexing program 3...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 3
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 3
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 3
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 3
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 3
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - SPACE [ ] found on line 3
DEBUG Lexer - SPACE [ ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 3
DEBUG Lexer - EOP [ $ ] found on line 3
INFO Lexer - Lex completed with 0 errors

INFO Lexer - Lexing program 4...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 4
DEBUG Lexer - SPACE [ ] found on line 4
DEBUG Lexer - SPACE [ ] found on line 4
DEBUG Lexer - I_TYPE [ int ] found on line 4
DEBUG Lexer - SPACE [ ] found on line 4
ERROR Lexer - Error: line 4 Unrecognized Token: @ Please reference grammar guide for this language.
Perhaps you tried to type an invalid character outside of the letters a through z?
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 4
DEBUG Lexer - EOP [ $ ] found on line 4
Error Lexer - Lex failed with 1 error(s)

```

```
INFO Lexer - Lexing program 5...
DEBUG Lexer - SPACE [ ] found on line 5
DEBUG Lexer - I_TYPE [ int ] found on line 5
DEBUG Lexer - SPACE [ ] found on line 5
DEBUG Lexer - ID [ a ] found on line 5
DEBUG Lexer - SPACE [ ] found on line 6
DEBUG Lexer - ID [ a ] found on line 6
DEBUG Lexer - SPACE [ ] found on line 6
DEBUG Lexer - ASSIGN_OP [ = ] found on line 6
DEBUG Lexer - SPACE [ ] found on line 6
DEBUG Lexer - ID [ a ] found on line 6
DEBUG Lexer - SPACE [ ] found on line 7
DEBUG Lexer - I_TYPE [ string ] found on line 7
DEBUG Lexer - SPACE [ ] found on line 7
DEBUG Lexer - ID [ b ] found on line 7
DEBUG Lexer - SPACE [ ] found on line 8
DEBUG Lexer - ID [ a ] found on line 8
DEBUG Lexer - SPACE [ ] found on line 8
DEBUG Lexer - ASSIGN_OP [ = ] found on line 8
DEBUG Lexer - SPACE [ ] found on line 8
DEBUG Lexer - ID [ b ] found on line 8
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 9
DEBUG Lexer - EOP [ $ ] found on line 9
INFO Lexer - Lex completed with 0 errors
```

Test Case # 5:

```

/*LongTestCase-
EverythingExceptBooleanDeclaration*/{/*IntDeclarati
on*/int a; b=0; /*WhileLoop*/while(a!=3)
{print(a)while(b!=3){print(b)b=1+bif(b==2)
{/*PrintStatement*/print("there is no
spoon")/*Thiswillidonothing*/}}b=0a=1+a}}$

```

```

INFO Lexer - Lexing program 1...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - I_TYPE [ int ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - I_TYPE [ int ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - ASSIGN_OP [ = ] found on line 1
DEBUG Lexer - digit [ 0 ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - ASSIGN_OP [ = ] found on line 1
DEBUG Lexer - digit [ 0 ] found on line 1
DEBUG Lexer - WhileStatement [ while ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - BOOL_INEQUAL [ != ] found on line 1
DEBUG Lexer - digit [ 3 ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - PRINT_START [ print ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - WhileStatement [ while ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - BOOL_INEQUAL [ != ] found on line 1
DEBUG Lexer - digit [ 3 ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - PRINT_START [ print ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - ASSIGN_OP [ = ] found on line 1
DEBUG Lexer - digit [ 1 ] found on line 1
DEBUG Lexer - intop [ + ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - IfStatement [ if ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - BOOL_EQUAL [ == ] found on line 1
DEBUG Lexer - digit [ 2 ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - PRINT_START [ print ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - StringExpr [ " ] found on line 1
DEBUG Lexer - ID [ t ] found on line 1
DEBUG Lexer - ID [ h ] found on line 1
DEBUG Lexer - ID [ e ] found on line 1
DEBUG Lexer - ID [ r ] found on line 1
INFO Lexer - ID [ e ] found on line 1

```

```
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ i ] found on line 1
DEBUG Lexer - ID [ s ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ n ] found on line 1
DEBUG Lexer - ID [ o ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ s ] found on line 1
DEBUG Lexer - ID [ p ] found on line 1
DEBUG Lexer - ID [ o ] found on line 1
DEBUG Lexer - ID [ o ] found on line 1
DEBUG Lexer - ID [ n ] found on line 1
DEBUG Lexer - StringExpr [ " ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 1
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 1
DEBUG Lexer - ID [ b ] found on line 1
DEBUG Lexer - ASSIGN_OP [ = ] found on line 1
DEBUG Lexer - digit [ 0 ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - ASSIGN_OP [ = ] found on line 1
DEBUG Lexer - digit [ 1 ] found on line 1
DEBUG Lexer - intop [ + ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 1
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 1
DEBUG Lexer - EOP [ $ ] found on line 1
INFO Lexer - Lex completed with 0 errors
```

Test Case # 6:

```

{ /* Comment!!! */ ( )
print=whileif"teststring"
intstringbooleanfalse
true
==!+= a 0123456789}$

```

```

INFO Lexer - Lexing program 1...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - StartParen [ ( ] found on line 1
DEBUG Lexer - CloseParen [ ) ] found on line 1
DEBUG Lexer - PRINT [ print ] found on line 2
DEBUG Lexer - ASSIGN_OP [ = ] found on line 2
DEBUG Lexer - WhileStatement [ while ] found on
line 2
DEBUG Lexer - IfStatement [ if ] found on line 2
DEBUG Lexer - StringExpr [ " ] found on line 2
DEBUG Lexer - ID [ t ] found on line 2
DEBUG Lexer - ID [ e ] found on line 2
DEBUG Lexer - ID [ s ] found on line 2
DEBUG Lexer - ID [ t ] found on line 2
DEBUG Lexer - I_TYPE [ string ] found on line 2
DEBUG Lexer - StringExpr [ " ] found on line 2
DEBUG Lexer - I_TYPE [ int ] found on line 3
DEBUG Lexer - I_TYPE [ string ] found on line 3
DEBUG Lexer - I_TYPE [ boolean ] found on line 3
DEBUG Lexer - boolval_F [ false ] found on line 3
DEBUG Lexer - boolval_T [ true ] found on line 4
DEBUG Lexer - BOOL_EQUAL [ == ] found on line 5
DEBUG Lexer - BOOL_INEQUAL [ != ] found on line 5
DEBUG Lexer - intop [ + ] found on line 5
DEBUG Lexer - SPACE [ ] found on line 5
DEBUG Lexer - ID [ a ] found on line 5
DEBUG Lexer - SPACE [ ] found on line 5
DEBUG Lexer - digit [ 0 ] found on line 5
DEBUG Lexer - digit [ 1 ] found on line 5
DEBUG Lexer - digit [ 2 ] found on line 5
DEBUG Lexer - digit [ 3 ] found on line 5
DEBUG Lexer - digit [ 4 ] found on line 5
DEBUG Lexer - digit [ 5 ] found on line 5
DEBUG Lexer - digit [ 6 ] found on line 5
DEBUG Lexer - digit [ 7 ] found on line 5
DEBUG Lexer - digit [ 8 ] found on line 5
DEBUG Lexer - digit [ 9 ] found on line 5
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 5
DEBUG Lexer - EOP [ $ ] found on line 5
INFO Lexer - Lex completed with 0 errors

```

## Test Case # 7:

```

{ /*/*This is */ "a test" *//*}$

```

```

INFO Lexer - Lexing program 1...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - StringExpr [ " ] found on line 1
DEBUG Lexer - ID [ a ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
DEBUG Lexer - ID [ t ] found on line 1
DEBUG Lexer - ID [ e ] found on line 1
DEBUG Lexer - ID [ s ] found on line 1
DEBUG Lexer - ID [ t ] found on line 1
DEBUG Lexer - StringExpr [ " ] found on line 1
DEBUG Lexer - SPACE [ ] found on line 1
ERROR Lexer - Error: line 1 Unrecognized Token: *
Please reference grammar guide for this language.
Perhaps you tried to type an invalid character
outside of the letters a through z?
ERROR Lexer - Error: line 1 Unrecognized Token: /
Please reference grammar guide for this language.
Perhaps you tried to type an invalid character
outside of the letters a through z?
ERROR Lexer - Error: Unterminated comment starting
on line 2. Lexing terminated due to fatal error.
Please add '*/' to end your comment.
Error Lexer - Lex failed with 3 error(s)

```

## Test Case # 8:

```

/*Long Test Case - Everything Except Boolean
Declaration */
{
/* Int Declaration */
int a
int b
a = 0
b=0
/* While Loop */
while (a != 3) {
print(a)
while (b != 3) {
print(b)
b = 1 + b
if (b == 2) {
/* Print Statement */
print("there is no spoon" /* This will do nothing
*/ )
}
}
b = 0
a = 1+a
}
}
$

```

Input:

 Compile

Output:

```

INFO Lexer - Lexing program 1...
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 2
DEBUG Lexer - I_TYPE [ int ] found on line 4
DEBUG Lexer - SPACE [ ] found on line 4
DEBUG Lexer - ID [ a ] found on line 4
DEBUG Lexer - I_TYPE [ int ] found on line 5
DEBUG Lexer - SPACE [ ] found on line 5
DEBUG Lexer - ID [ b ] found on line 5
DEBUG Lexer - ID [ a ] found on line 6
DEBUG Lexer - SPACE [ ] found on line 6
DEBUG Lexer - ASSIGN_OP [ = ] found on line 6
DEBUG Lexer - SPACE [ ] found on line 6
DEBUG Lexer - digit [ 0 ] found on line 6
DEBUG Lexer - ID [ b ] found on line 7
DEBUG Lexer - ASSIGN_OP [ = ] found on line 7
DEBUG Lexer - digit [ 0 ] found on line 7
DEBUG Lexer - WhileStatement [ while ] found on line 9
DEBUG Lexer - SPACE [ ] found on line 9
DEBUG Lexer - StartParen [ ( ] found on line 9
DEBUG Lexer - ID [ a ] found on line 9
DEBUG Lexer - SPACE [ ] found on line 9
DEBUG Lexer - BOOL_INEQUAL [ != ] found on line 9
DEBUG Lexer - SPACE [ ] found on line 9
DEBUG Lexer - digit [ 3 ] found on line 9
DEBUG Lexer - CloseParen [ ) ] found on line 9
DEBUG Lexer - SPACE [ ] found on line 9
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 9
DEBUG Lexer - PRINT [ print ] found on line 10
DEBUG Lexer - StartParen [ ( ] found on line 10
DEBUG Lexer - ID [ a ] found on line 10
DEBUG Lexer - CloseParen [ ) ] found on line 10
DEBUG Lexer - WhileStatement [ while ] found on line 11
DEBUG Lexer - SPACE [ ] found on line 11
DEBUG Lexer - StartParen [ ( ] found on line 11
DEBUG Lexer - ID [ b ] found on line 11
DEBUG Lexer - SPACE [ ] found on line 11
DEBUG Lexer - BOOL_INEQUAL [ != ] found on line 11
DEBUG Lexer - SPACE [ ] found on line 11
DEBUG Lexer - digit [ 3 ] found on line 11
DEBUG Lexer - CloseParen [ ) ] found on line 11
DEBUG Lexer - SPACE [ ] found on line 11
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 11
DEBUG Lexer - PRINT [ print ] found on line 12
DEBUG Lexer - StartParen [ ( ] found on line 12
DEBUG Lexer - ID [ b ] found on line 12
DEBUG Lexer - CloseParen [ ) ] found on line 12
DEBUG Lexer - ID [ b ] found on line 13
DEBUG Lexer - SPACE [ ] found on line 13
DEBUG Lexer - ASSIGN_OP [ = ] found on line 13
DEBUG Lexer - SPACE [ ] found on line 13
DEBUG Lexer - digit [ 1 ] found on line 13
DEBUG Lexer - SPACE [ ] found on line 13
DEBUG Lexer - intop [ + ] found on line 13
DEBUG Lexer - SPACE [ ] found on line 13
DEBUG Lexer - ID [ b ] found on line 13
DEBUG Lexer - IfStatement [ if ] found on line 14
DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - StartParen [ ( ] found on line 14
DEBUG Lexer - ID [ b ] found on line 14
DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - BOOL_EQUAL [ == ] found on line 14
DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - digit [ 2 ] found on line 14

```

```

DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - StartParen [ ( ] found on line 14
DEBUG Lexer - ID [ b ] found on line 14
DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - BOOL_EQUAL [ == ] found on line 14
DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - digit [ 2 ] found on line 14
DEBUG Lexer - CloseParen [ ) ] found on line 14
DEBUG Lexer - SPACE [ ] found on line 14
DEBUG Lexer - OPEN_BLOCK [ { ] found on line 14
DEBUG Lexer - PRINT [ print ] found on line 16
DEBUG Lexer - StartParen [ ( ] found on line 16
DEBUG Lexer - StringExpr [ " ] found on line 16
DEBUG Lexer - ID [ t ] found on line 16
DEBUG Lexer - ID [ h ] found on line 16
DEBUG Lexer - ID [ e ] found on line 16
DEBUG Lexer - ID [ r ] found on line 16
DEBUG Lexer - ID [ e ] found on line 16
DEBUG Lexer - SPACE [ ] found on line 16
DEBUG Lexer - ID [ i ] found on line 16
DEBUG Lexer - ID [ s ] found on line 16
DEBUG Lexer - SPACE [ ] found on line 16
DEBUG Lexer - ID [ n ] found on line 16
DEBUG Lexer - ID [ o ] found on line 16
DEBUG Lexer - SPACE [ ] found on line 16
DEBUG Lexer - ID [ s ] found on line 16
DEBUG Lexer - ID [ p ] found on line 16
DEBUG Lexer - ID [ o ] found on line 16
DEBUG Lexer - ID [ o ] found on line 16
DEBUG Lexer - ID [ n ] found on line 16
DEBUG Lexer - StringExpr [ " ] found on line 16
DEBUG Lexer - SPACE [ ] found on line 16
DEBUG Lexer - SPACE [ ] found on line 16
DEBUG Lexer - CloseParen [ ) ] found on line 16
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 17
DEBUG Lexer - CLOSE_BLOCK [ } ] found on line 18
DEBUG Lexer - ID [ b ] found on line 19
DEBUG Lexer - SPACE [ ] found on line 19
DEBUG Lexer - ASSIGN_OP [ = ] found on line 19
DEBUG Lexer - SPACE [ ] found on line 19
DEBUG Lexer - digit [ 0 ] found on line 19
DEBUG Lexer - ID [ a ] found on line 20
DEBUG Lexer - SPACE [ ] found on line 20
DEBUG Lexer - ASSIGN_OP [ = ] found on line 20
DEBUG Lexer - SPACE [ ] found on line 20

```

Overall, all errors and warnings are recognized and described in detail with recommended solutions. Throughout testing I found bugs and things that I wanted to change so the testing was overall very useful! The test results above are the final results after making these last adjustments.