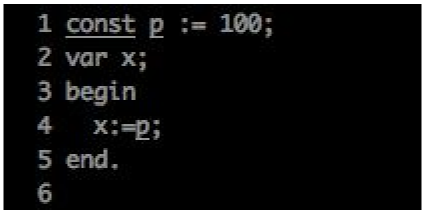
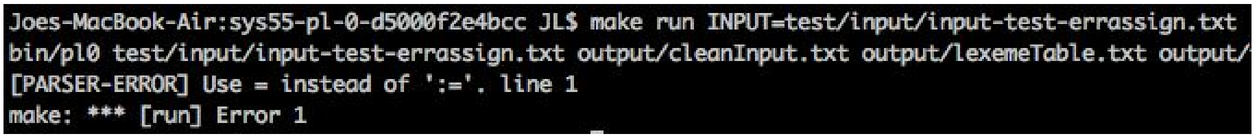
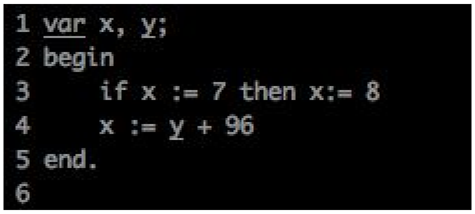


```
1 const p := 100;
2 var x;
3 begin
4   x:=p;
5 end.
6
```

1. **input-test-errassign.txt**

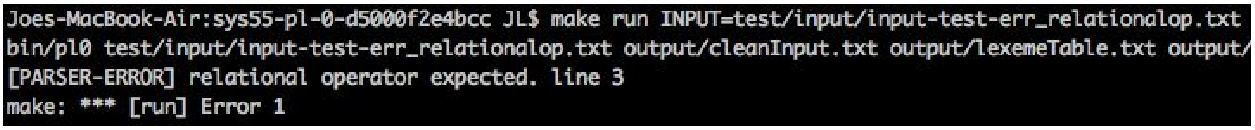


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errassign.txt
bin/pl0 test/input/input-test-errassign.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] Use = instead of ':='. line 1
make: *** [run] Error 1
```

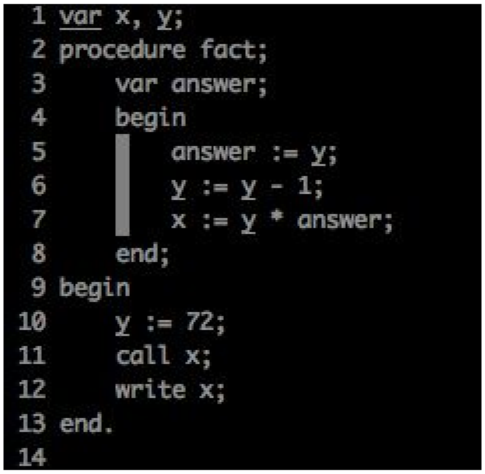


```
1 var x, y;
2 begin
3     if x := 7 then x:= 8
4     x := y + 96
5 end.
6
```

2. **input-test-err_relationalop.txt**

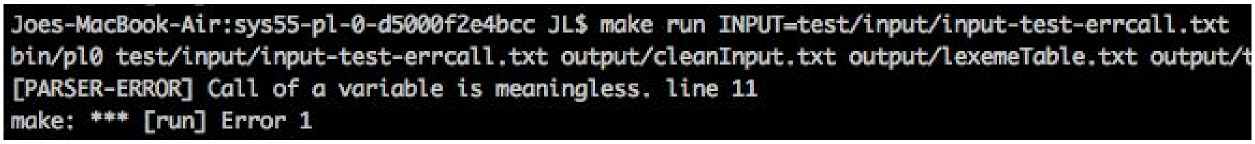


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-err_relationalop.txt
bin/pl0 test/input/input-test-err_relationalop.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] relational operator expected. line 3
make: *** [run] Error 1
```

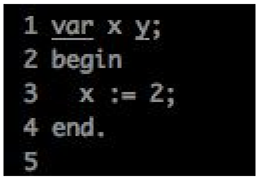


```
 1 var x, y;
 2 procedure fact;
 3     var answer;
 4     begin
 5         answer := y;
 6         y := y - 1;
 7         x := y * answer;
 8     end;
 9 begin
10     y := 72;
11     call x;
12     write x;
13 end.
14
```

3. **input-test-errcall.txt**

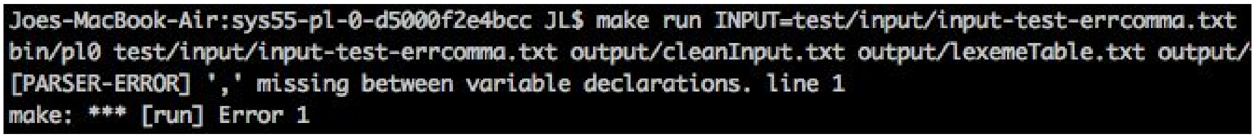


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errcall.txt
bin/pl0 test/input/input-test-errcall.txt output/cleanInput.txt output/lexemeTable.txt output/t
[PARSER-ERROR] Call of a variable is meaningless. line 11
make: *** [run] Error 1
```

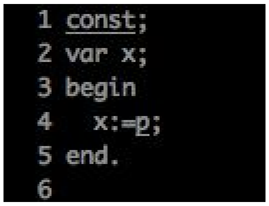


```
1 var x y;
2 begin
3   x := 2;
4 end.
5
```

4. input-test-errcomma.txt

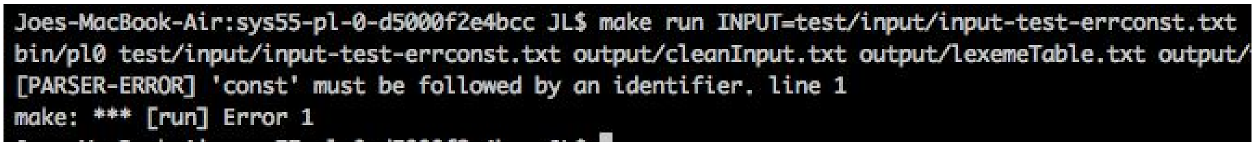


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errcomma.txt
bin/pl0 test/input/input-test-errcomma.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] ',' missing between variable declarations. line 1
make: *** [run] Error 1
```

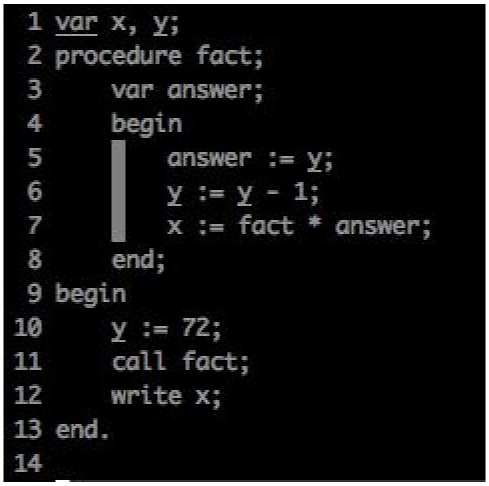


```
1 const;
2 var x;
3 begin
4   x:=p;
5 end.
6
```

5. input-test-errconst.txt

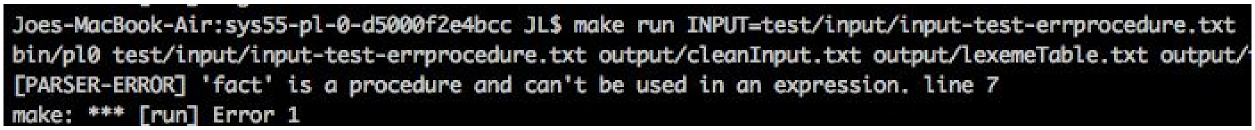


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errconst.txt
bin/pl0 test/input/input-test-errconst.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] 'const' must be followed by an identifier. line 1
make: *** [run] Error 1
```



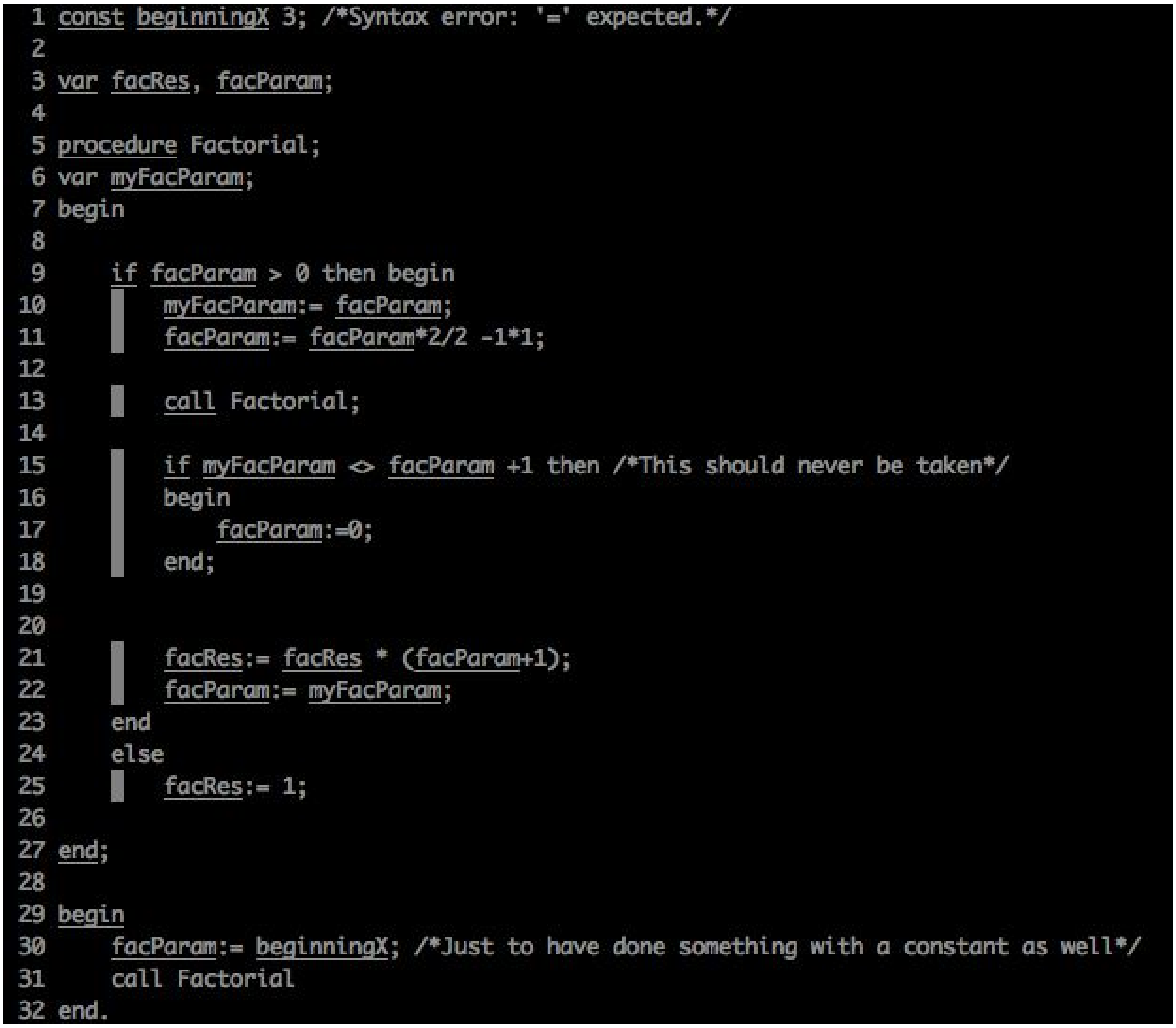
```
 1 var x, y;
 2 procedure fact;
 3     var answer;
 4     begin
 5         answer := y;
 6         y := y - 1;
 7         x := fact * answer;
 8     end;
 9 begin
10     y := 72;
11     call fact;
12     write x;
13 end.
14
```

6. input-test-errprocedure.txt

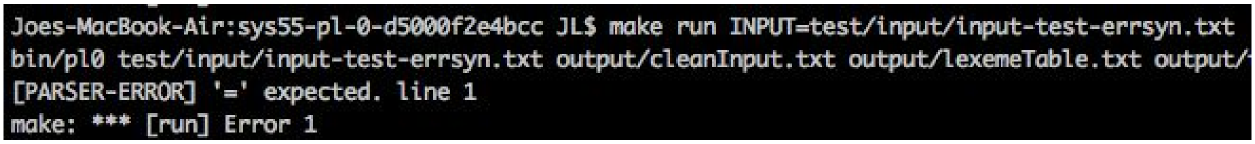


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errprocedure.txt
bin/pl0 test/input/input-test-errprocedure.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] 'fact' is a procedure and can't be used in an expression. line 7
make: *** [run] Error 1
```

7. input-test-errsyn.txt

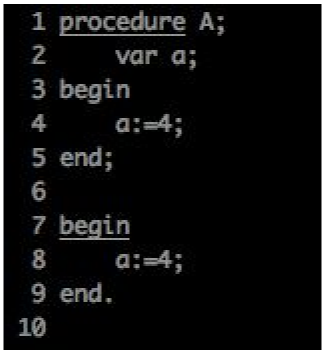


```
 1 const beginningX 3; /*Syntax error: '=' expected.*/
 2
 3 var facRes, facParam;
 4
 5 procedure Factorial;
 6 var myFacParam;
 7 begin
 8
 9     if facParam > 0 then begin
10         myFacParam:= facParam;
11         facParam:= facParam*2/2 -1*1;
12
13         call Factorial;
14
15         if myFacParam <> facParam +1 then /*This should never be taken*/
16         begin
17             facParam:=0;
18         end;
19
20
21         facRes:= facRes * (facParam+1);
22         facParam:= myFacParam;
23     end
24     else
25         facRes:= 1;
26
27 end;
28
29 begin
30     facParam:= beginningX; /*Just to have done something with a constant as well*/
31     call Factorial
32 end.
```

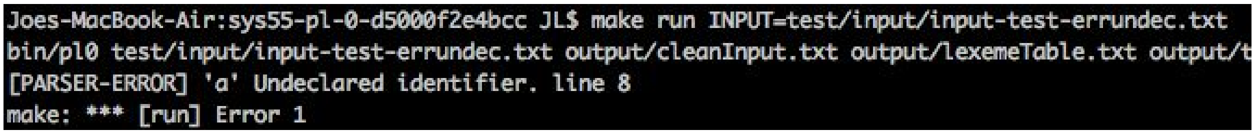


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errsyn.txt
bin/pl0 test/input/input-test-errsyn.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] '=' expected. line 1
make: *** [run] Error 1
```

8. input-test-errundec.txt

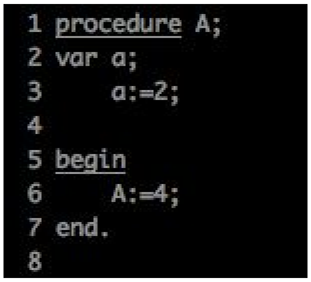


```
 1 procedure A;
 2     var a;
 3 begin
 4     a:=4;
 5 end;
 6
 7 begin
 8     a:=4;
 9 end.
10
```

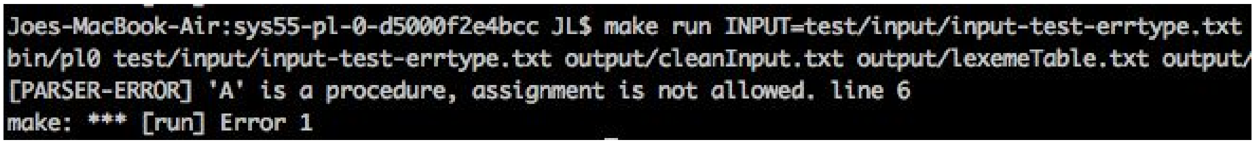


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errundec.txt
bin/pl0 test/input/input-test-errundec.txt output/cleanInput.txt output/lexemeTable.txt output/t
[PARSER-ERROR] 'a' Undeclared identifier. line 8
make: *** [run] Error 1
```

9. input-test-errtype.txt

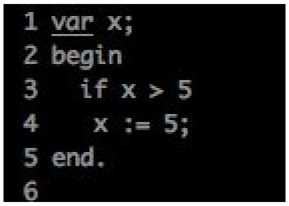


```
1 procedure A;
2 var a;
3     a:=2;
4
5 begin
6     A:=4;
7 end.
8
```

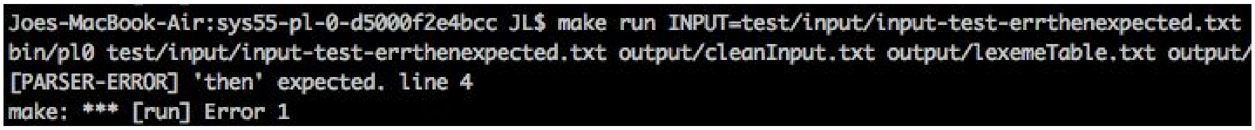


```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errtype.txt
bin/pl0 test/input/input-test-errtype.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] 'A' is a procedure, assignment is not allowed. line 6
make: *** [run] Error 1
```

10. input-test-errthenexpected.txt



```
1 var x;
2 begin
3   if x > 5
4     x := 5;
5 end.
6
```



```
Joes-MacBook-Air:sys55-pl-0-d5000f2e4bcc JL$ make run INPUT=test/input/input-test-errthenexpected.txt
bin/pl0 test/input/input-test-errthenexpected.txt output/cleanInput.txt output/lexemeTable.txt output/
[PARSER-ERROR] 'then' expected. line 4
make: *** [run] Error 1
```