

## PL/0 Program Errors/ Do-Not List

This is a guide to 10 common program errors and their solutions.

1. Not ending the program in a period.

Example:

```
var x, y;
```

```
begin
```

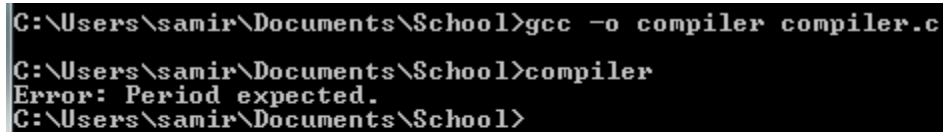
```
  y := 3;
```

```
  x := y + 56;
```

```
end
```

<- Here you can see there is a period missing. This does not follow the grammar.

Resulting Output:



```
C:\Users\samir\Documents\School>gcc -o compiler compiler.c
C:\Users\samir\Documents\School>compiler
Error: Period expected.
C:\Users\samir\Documents\School>
```

Solution:

```
var x, y;
```

```
begin
```

```
  y := 3;
```

```
  x := y + 56;
```

```
end.
```

<- Add the period when ending the program. This way it will run through the compiler successfully.

2. Undeclared Identifier

Example:

```
var x, y;
```

```
begin
```

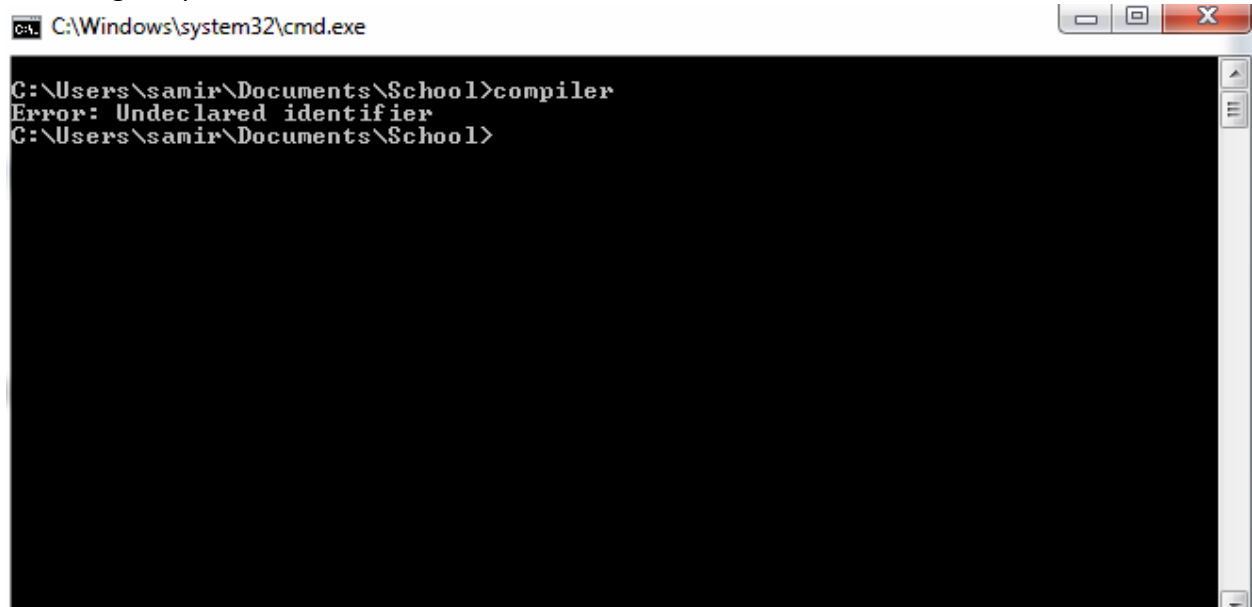
```
  z := 3;
```

<- Here you can see that z wasn't declared.

```
  x := y + 56;
```

```
end.
```

Resulting Output:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt shows the user is in the directory 'C:\Users\samir\Documents\School'. They entered the command 'compiler', which resulted in the error message 'Error: Undeclared identifier'. The prompt then returns to 'C:\Users\samir\Documents\School>'.

```
C:\Windows\system32\cmd.exe
C:\Users\samir\Documents\School>compiler
Error: Undeclared identifier
C:\Users\samir\Documents\School>
```

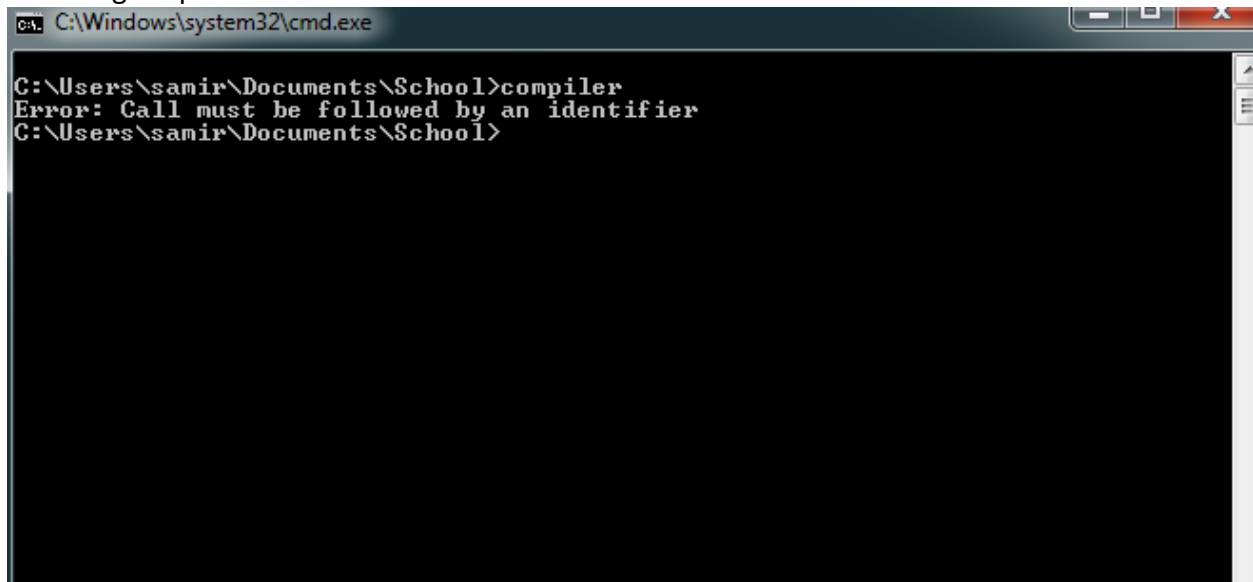
Solution: Declare the identifier just as you do with the other identifiers.

### 3. Call must be followed by an identifier

Example:

```
var f, n;
procedure fact;
  var ans1;
  begin
    ans1:=n;
    n:= n-1;
    if n = 0 then f := 1;
    if n > 0 then call fact;
    f:=f*ans1;
  end;
begin
  n:=3;
  call;          <- Call isn't followed by an identifier (fact, in this case)
  write f;
end.
```

Resulting output:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt shows the current directory as 'C:\Users\samir\Documents\School'. The user has entered the command 'compiler', which has resulted in an error message: 'Error: Call must be followed by an identifier'. The prompt is now waiting for the next command.

```
C:\Windows\system32\cmd.exe
C:\Users\samir\Documents\School>compiler
Error: Call must be followed by an identifier
C:\Users\samir\Documents\School>
```

Solution: Change the indicated line to:

```
call fact;
```

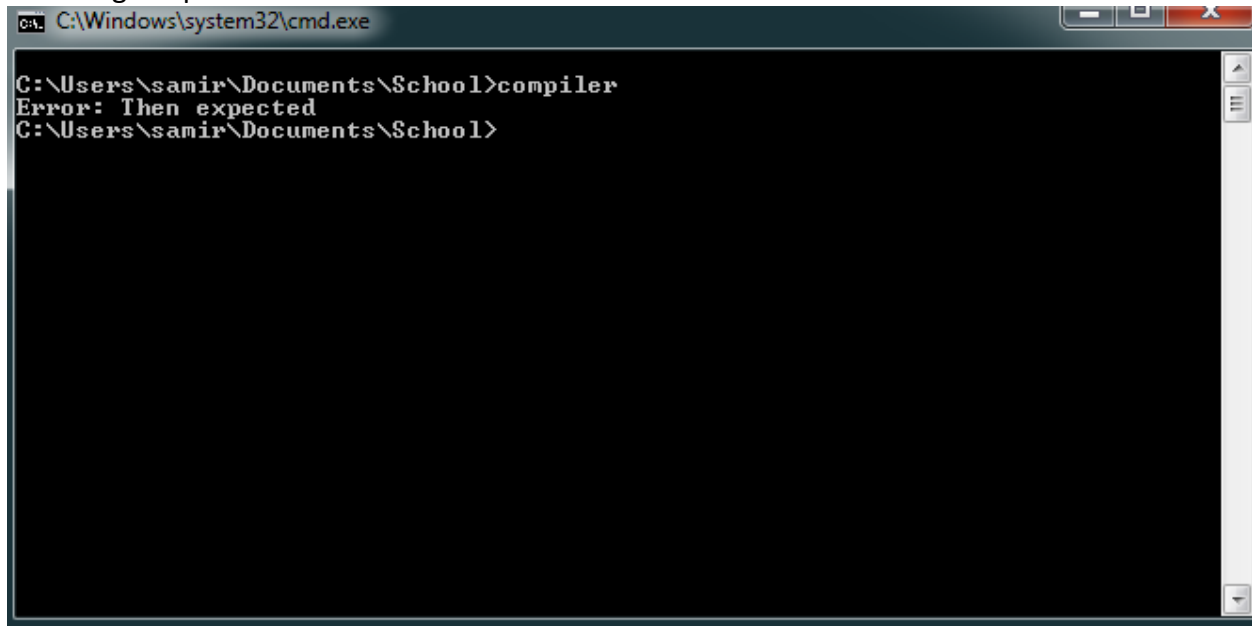
This will make sure there is an actual procedure being called.

4. Then expected

Example:

```
var f, n;
procedure fact;
  var ans1;
  begin
    ans1:=n;
    n:= n-1;
    if n = 0 f := 1; <- A then is expected here. If's only work with then's in PL/0.
    if n > 0 then call fact;
    f:=f*ans1;
  end;
begin
  n:=3;
  call fact;
  write f;
end.
```

Resulting Output:



```
C:\Windows\system32\cmd.exe

C:\Users\samir\Documents\School>compiler
Error: Then expected
C:\Users\samir\Documents\School>
```

Solution: Add a “then” with an expression to follow.

5. Do expected

Example:

var x, y;

begin

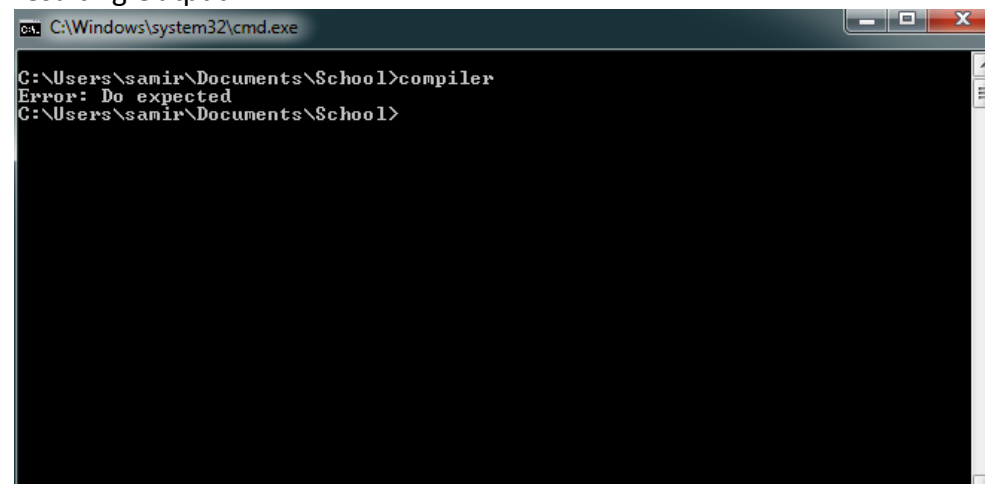
    y := 3;

    while x>3 <- there is a missing “do” to execute the following line

    x := y + 56;

end.

Resulting Output:



```
C:\Windows\system32\cmd.exe

C:\Users\samir\Documents\School>compiler
Error: Do expected
C:\Users\samir\Documents\School>
```

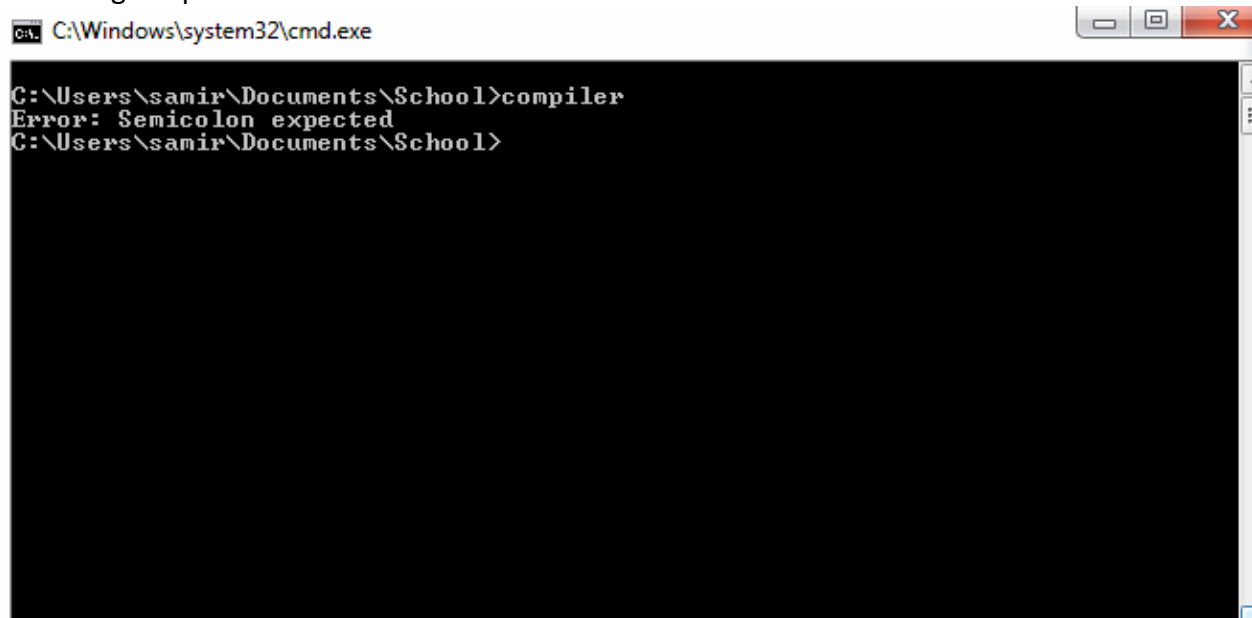
Solution: Add a “do” along with a line to go with it. In conjunction with the “while”, this will form a loop that will execute lines of code while a condition is met.

## 6. Semicolon Expected

Example:

```
var x, y;  
begin  
    y := 3 <- There should be a semicolon here  
    x := y + 56;  
end.
```

Resulting Output:



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The prompt is at "C:\Users\samir\Documents\School>". The user has entered the command "compiler", and the output is "Error: Semicolon expected". The prompt is now "C:\Users\samir\Documents\School>".

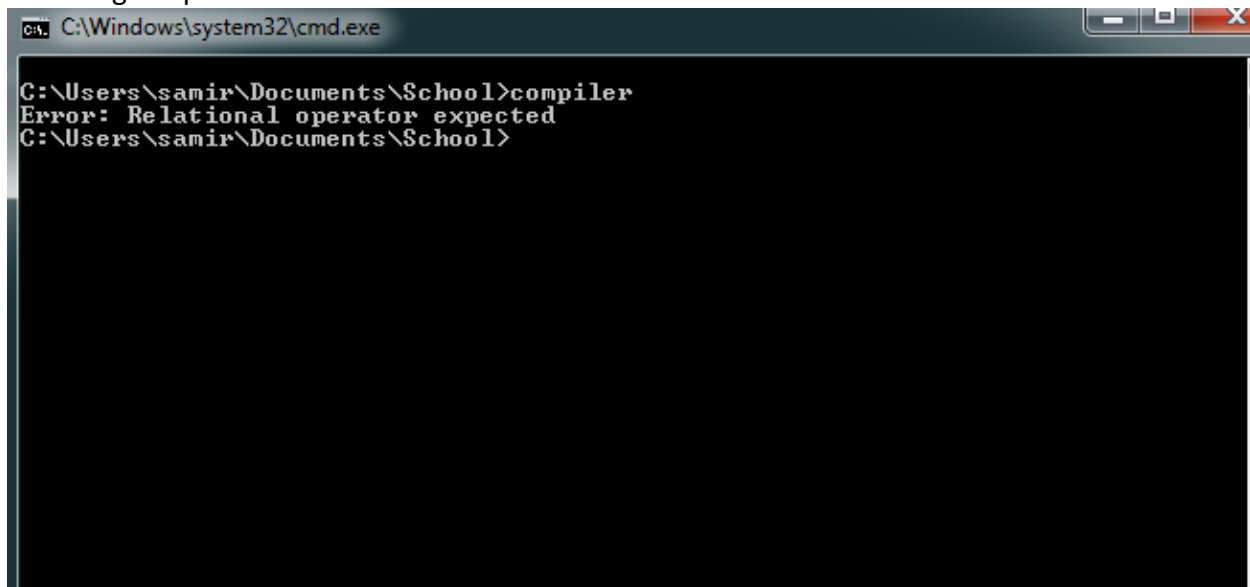
Solution: include a semicolon at the end of the specified line.

## 7. Relational operator expected

Example:

```
var x, y;  
begin  
    if x :=3 then x:=4    <- := isn't a relational operator. It is the becomes symbol.  
    x := y + 56;  
end.
```

Resulting Output:

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The command prompt shows the following text:

```
C:\Users\samir\Documents\School>compiler
Error: Relational operator expected
C:\Users\samir\Documents\School>
```

Solution: Make sure you have a relational operator in the if statement. Refer to the user guide's relational operations for more info.

#### 8. Semicolon or Comma Missing

Example:

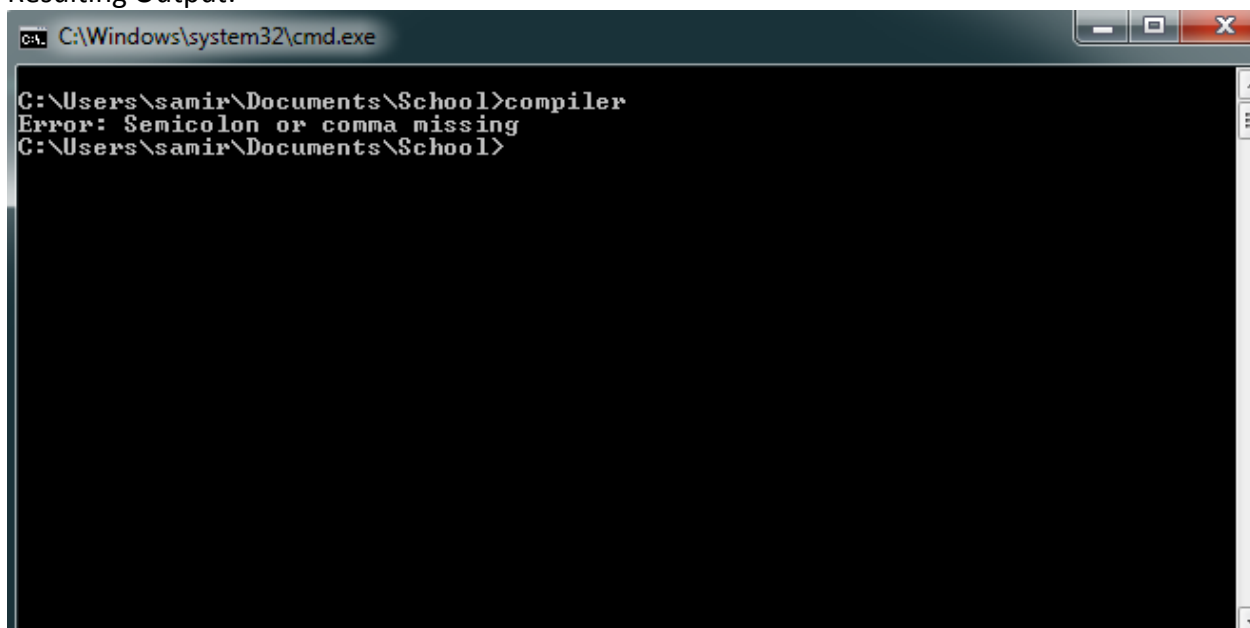
var x y; <-Comma is missing here

begin

    x := y + 56;

end.

Resulting Output:

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The command prompt shows the following text:

```
C:\Users\samir\Documents\School>compiler
Error: Semicolon or comma missing
C:\Users\samir\Documents\School>
```

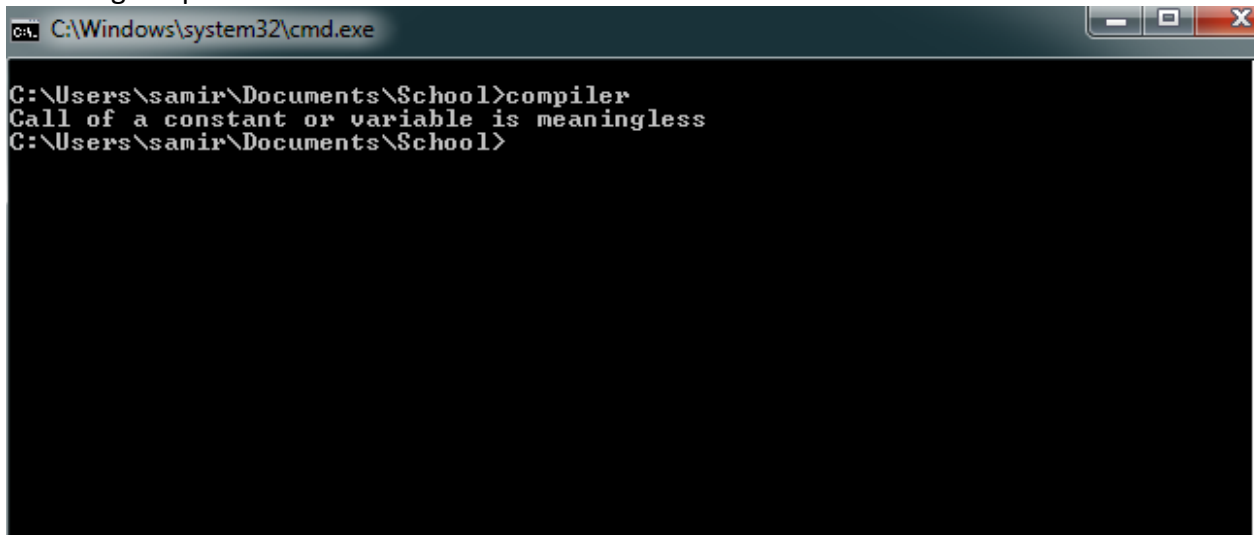
Solution: Include commas or semicolons as needed. Refer to the user guide for information on syntax.

9. Call of constant or variable is meaningless

Example:

```
var x, y;  
begin  
  y; <-This is a meaningless call of the variable y.  
  x := y + 56;  
end.
```

Resulting Output:



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The command prompt is open at the directory "C:\Users\samir\Documents\School". The user has entered the command "compiler", and the output displayed is "Call of a constant or variable is meaningless". The prompt then returns to "C:\Users\samir\Documents\School>".

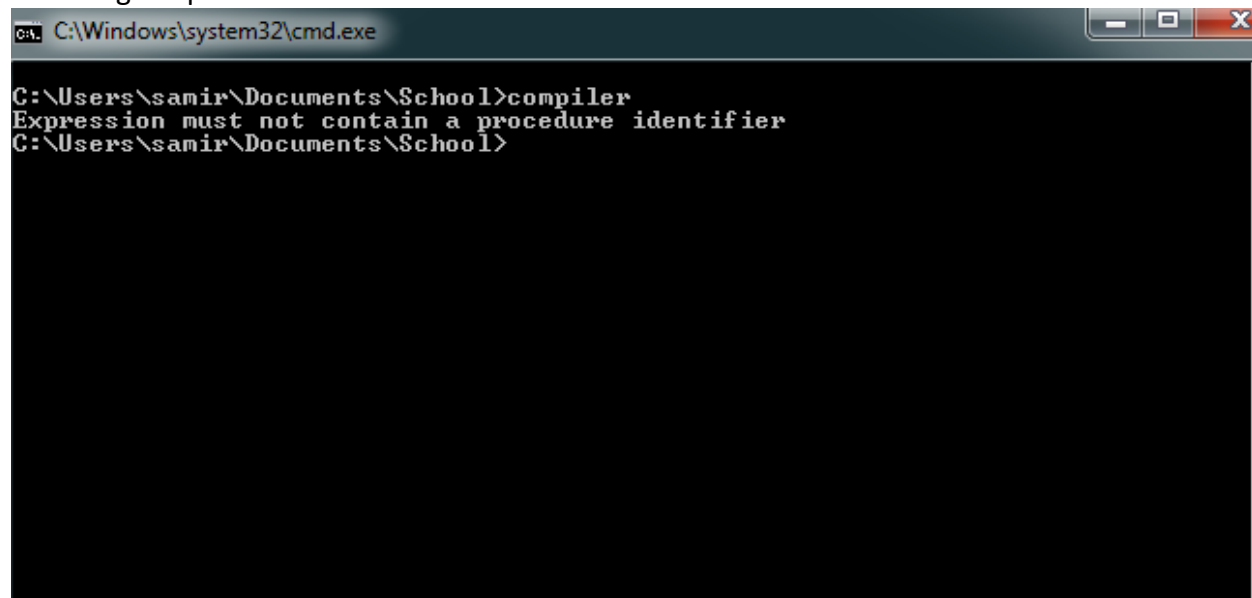
Solution: A good line would be something like `y :=3;` If you are having trouble using variables, refer to the User Guide included.

10. Expression must not include a procedure identifier

Example:

```
var f, n;  
procedure fact;  
  var ans1;  
  begin  
    ans1:=n;  
    n:= n-1;  
    f:=fact*ans1; <- fact is a procedure. It cannot be used here in this fashion.  
  end;  
begin  
  n:=3;  
  call fact;  
  write f;  
end.
```

Resulting Output:

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Windows\system32\cmd.exe". The command prompt shows the user's current directory as "C:\Users\samir\Documents\School". The user has entered the command "compiler", and the system has responded with the error message "Expression must not contain a procedure identifier". The prompt is now waiting for the next command.

```
C:\Windows\system32\cmd.exe
C:\Users\samir\Documents\School>compiler
Expression must not contain a procedure identifier
C:\Users\samir\Documents\School>
```

Solution: Do not use procedure names in these kinds of expressions. Refer to the user guide if you are confused about procedures or mathematical expressions.