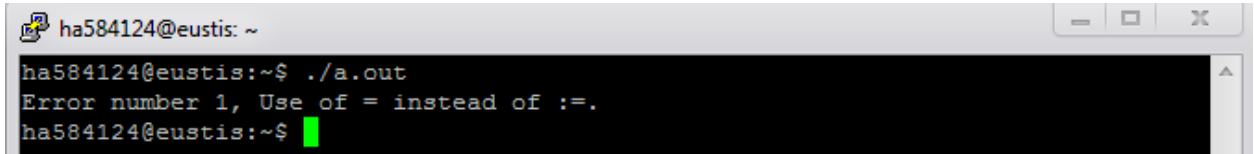# ERRORS

1. Use = instead of :=.

```
const a := 3;
var y;

begin
    y := a;
end.
```
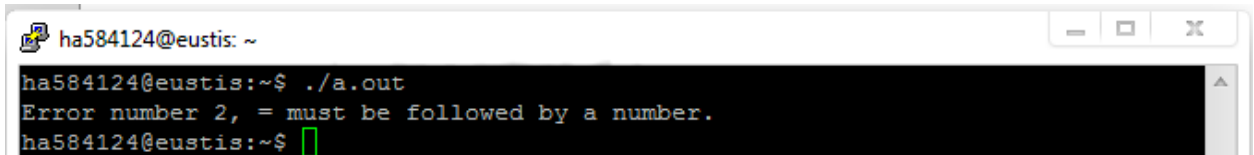
```
ha584124@eustis: ~                                    —  ☐  ✕

ha584124@eustis:~$ ./a.out
Error number 1, Use of = instead of :=.
ha584124@eustis:~$ █
```

2. = must be followed by a number.

```
const a = r;
var y;

begin
    y := a;
end.
```
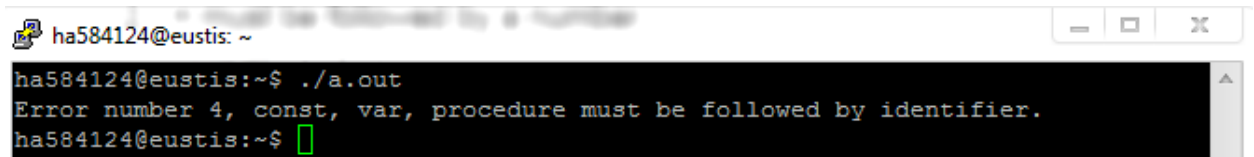
```
ha584124@eustis: ~                                    —  ☐  ✕

ha584124@eustis:~$ ./a.out
Error number 2, = must be followed by a number.
ha584124@eustis:~$ ▯
```

3. **const**, **var**, **procedure** must be followed by identifier.

```
const a = 3;
var;

begin
    y := a;
end.
```

```
ha584124@eustis: ~                                    —  ☐  ✕

ha584124@eustis:~$ ./a.out
Error number 4, const, var, procedure must be followed by identifier.
ha584124@eustis:~$ ▯
```

4. Semicolon or comma missing.
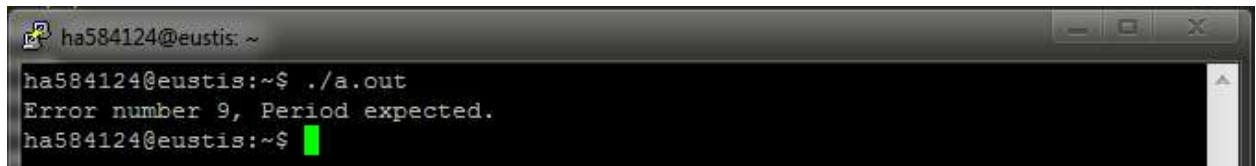
```
const a = 3
var x;

begin
    y := a;
end.
```

```
ha584124@eustis: ~
ha584124@eustis:~$ ./a.out
Error number 5, Semicolon or comma missing.
ha584124@eustis:~$
```

5. Period expected.

```
const a = 3;
var x;

begin
    x := a;
end
```
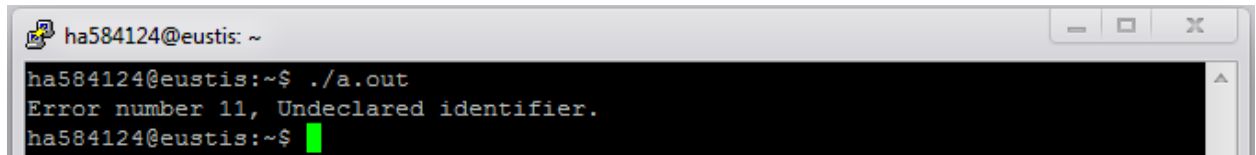
```
ha584124@eustis: ~
ha584124@eustis:~$ ./a.out
Error number 9, Period expected.
ha584124@eustis:~$
```

6. Undeclared identifier.

```
const a = 3;
var x;

begin
    y := a;
end.
```
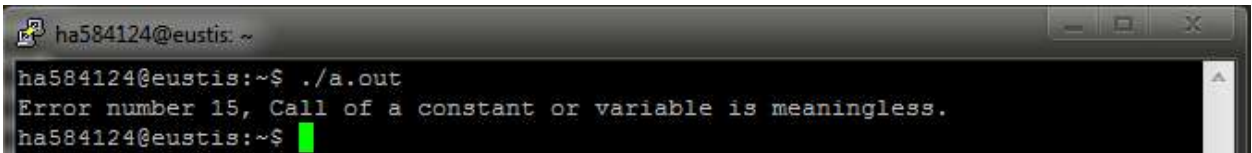
```
ha584124@eustis: ~
ha584124@eustis:~$ ./a.out
Error number 11, Undeclared identifier.
ha584124@eustis:~$
```

7. Call of a constant or variable is meaningless.

```
const a = 3;
var x;

begin
      call x;
      x := a;
end.
```

```
ha584124@eustis:~$ ./a.out
Error number 15, Call of a constant or variable is meaningless.
ha584124@eustis:~$
```

8. **then** expected.

```
const a = 3;
var x;

begin
      if x > 0
      x := a;
end.
```
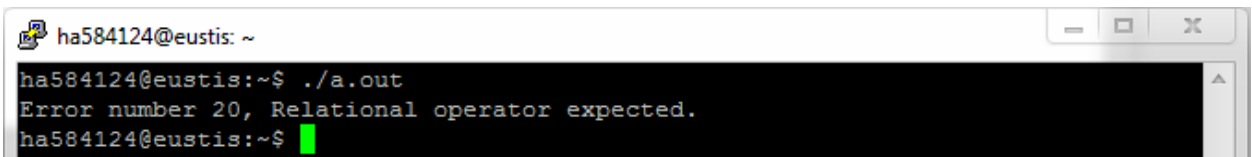
```
ha584124@eustis:~$ ./a.out
Error number 16, then expected.
ha584124@eustis:~$
```

9. Relational operator expected.

```
const a = 3;
var x;

begin
      if x + 0 then
      x := a;
end.
```
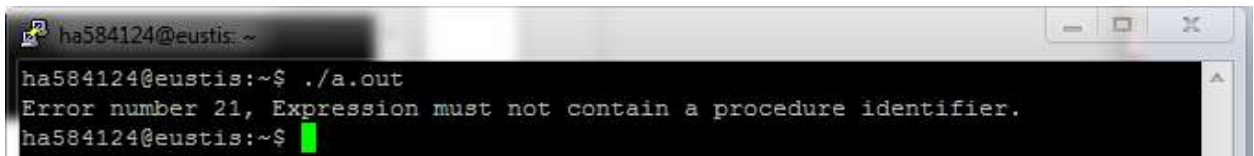
```
ha584124@eustis:~$ ./a.out
Error number 20, Relational operator expected.
ha584124@eustis:~$
```

10. Expression must not contain a procedure identifier.

```
const a = 3;
var x;
procedure test;

begin

        x := a + test;
end.
```

```
ha584124@eustis: ~                                        —  □  X
ha584124@eustis:~$ ./a.out
Error number 21, Expression must not contain a procedure identifier.
ha584124@eustis:~$
```
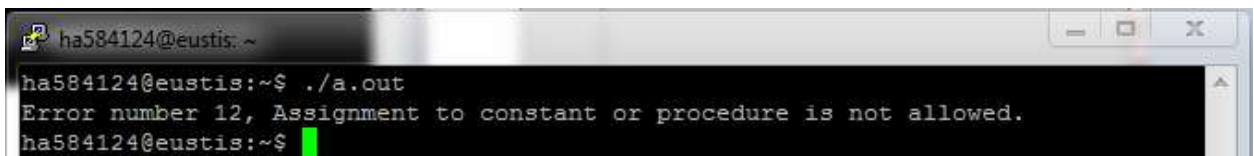
11. Assignment to constant or procedure is not allowed.

```
const a = 3;
var x;

begin

        a := 5 + x;
end.
```

```
ha584124@eustis: ~                                        —  □  X
ha584124@eustis:~$ ./a.out
Error number 12, Assignment to constant or procedure is not allowed.
ha584124@eustis:~$
```

12. An expression cannot begin with this symbol.

```
const a = 3;
var x;

begin

        x := const + a;
end.
```

```
ha584124@eustis: ~                                        —  □  X
ha584124@eustis:~$ ./a.out
Error number 24, An expression cannot begin with this symbol.
ha584124@eustis:~$
```

SUCCESFUL RUN

```
const beginningX= 3;

var facRes, facParam;

procedure Factorial;
var myFacParam;
begin

        if facParam > 0 then begin
                myFacParam:= facParam;
                facParam:= facParam*2/2 -1*1;

                call Factorial;

                if myFacParam <> facParam +1 then /*This should never be taken*/
                begin
                        facParam:=0;
                end;


                facRes:= facRes * (facParam+1);
                facParam:= myFacParam;
        end
        else
                facRes:= 1;

end;

begin
        facParam:= beginningX; /*Just to have done something with a constant as
well*/
        call Factorial
end.
```
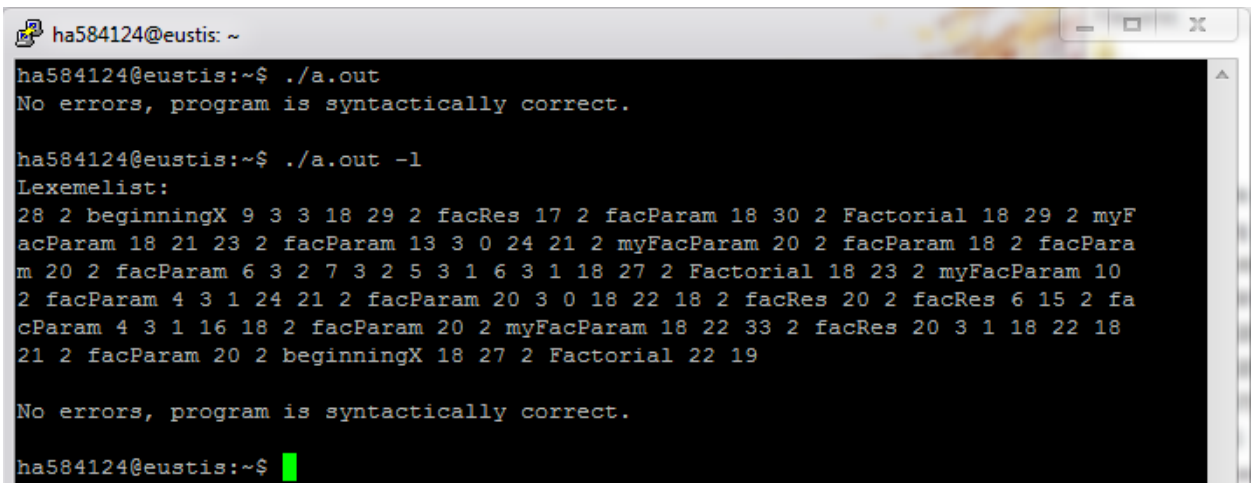
```
ha584124@eustis: ~                                               _ □ X

ha584124@eustis:~$ ./a.out
No errors, program is syntactically correct.

ha584124@eustis:~$ ./a.out -l
Lexemelist:
28 2 beginningX 9 3 3 18 29 2 facRes 17 2 facParam 18 30 2 Factorial 18 29 2 myF
acParam 18 21 23 2 facParam 13 3 0 24 21 2 myFacParam 20 2 facParam 18 2 facPara
m 20 2 facParam 6 3 2 7 3 2 5 3 1 6 3 1 18 27 2 Factorial 18 23 2 myFacParam 10
2 facParam 4 3 1 24 21 2 facParam 20 3 0 18 22 18 2 facRes 20 2 facRes 6 15 2 fa
cParam 4 3 1 16 18 2 facParam 20 2 myFacParam 18 22 33 2 facRes 20 3 1 18 22 18
21 2 facParam 20 2 beginningX 18 27 2 Factorial 22 19

No errors, program is syntactically correct.

ha584124@eustis:~$
```