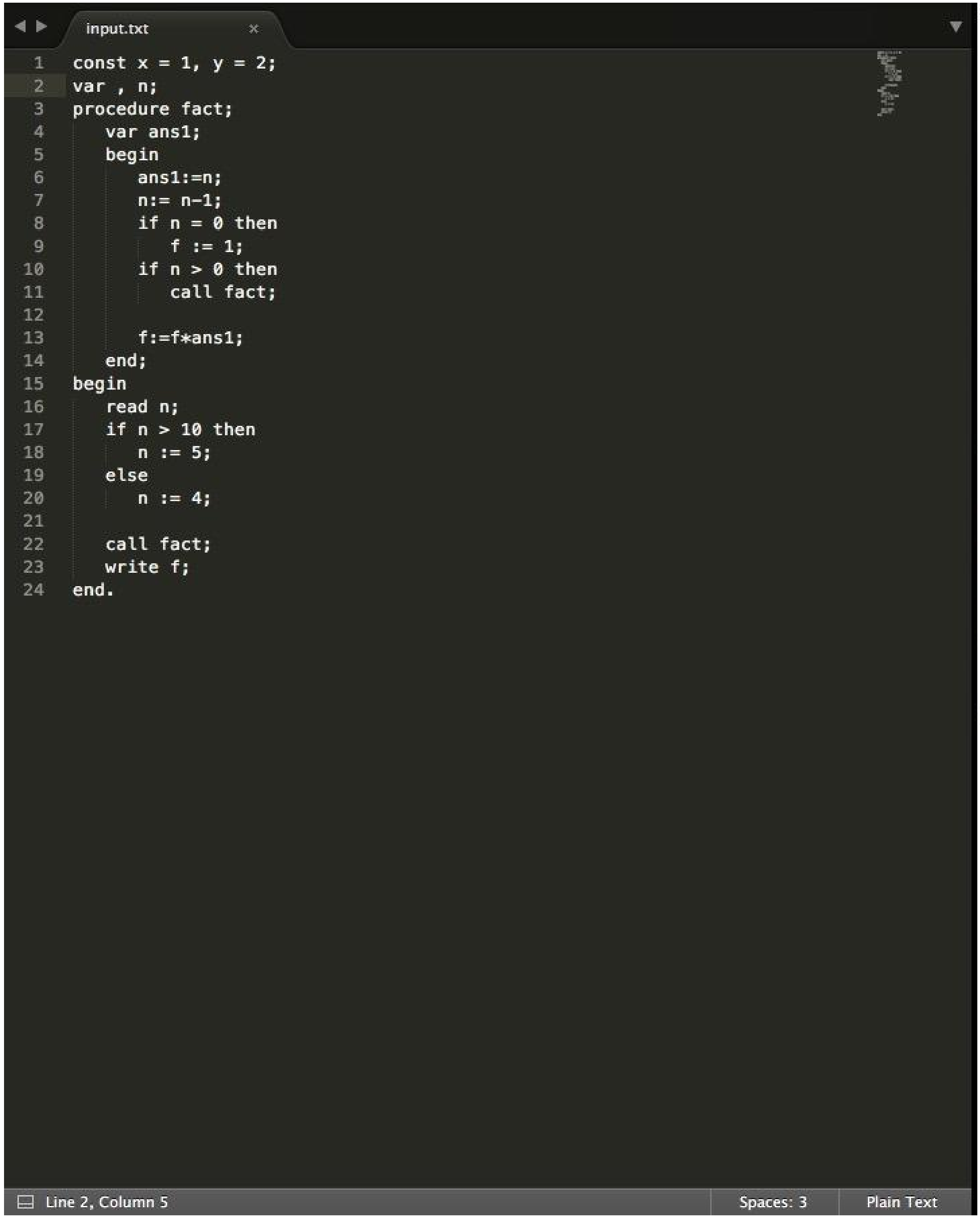
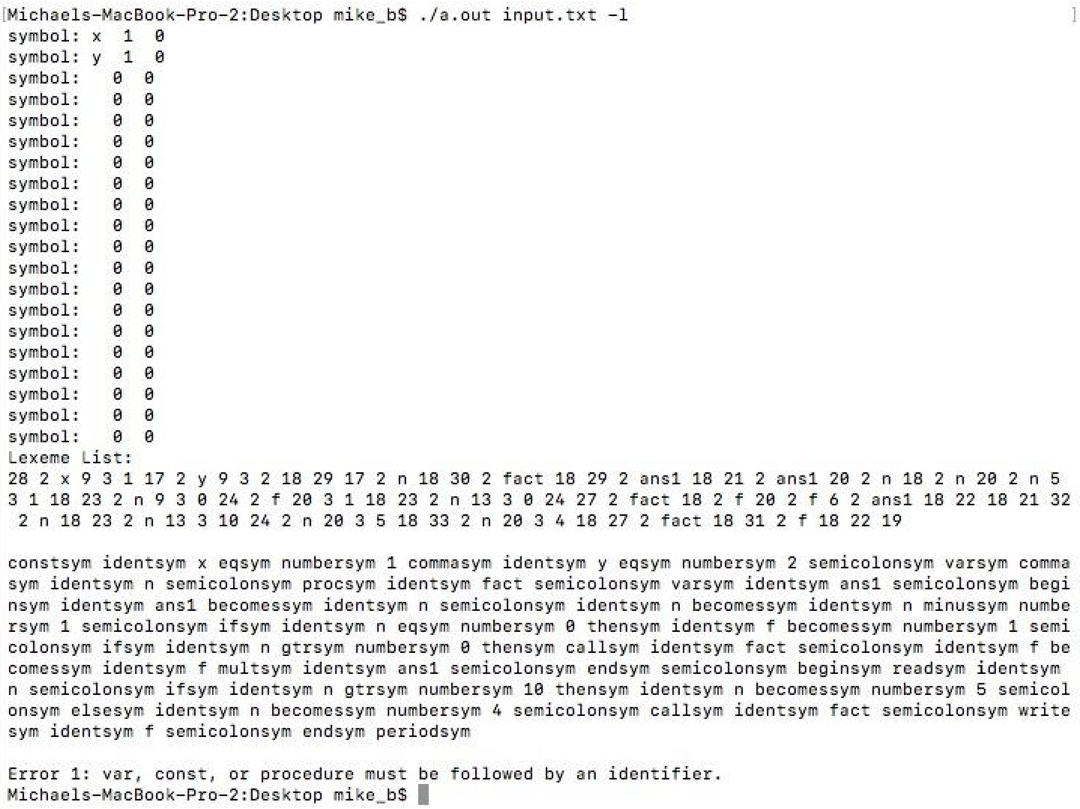
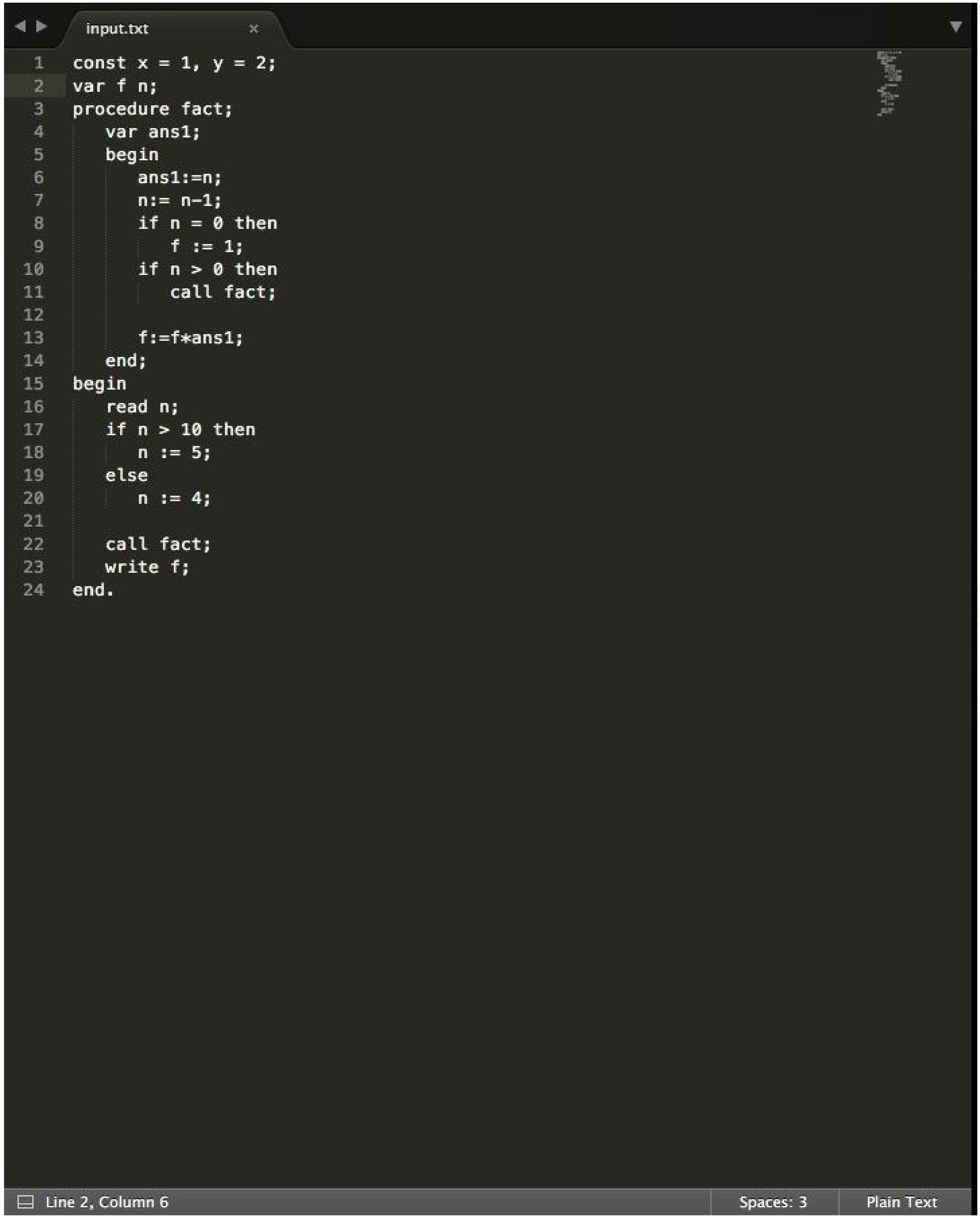


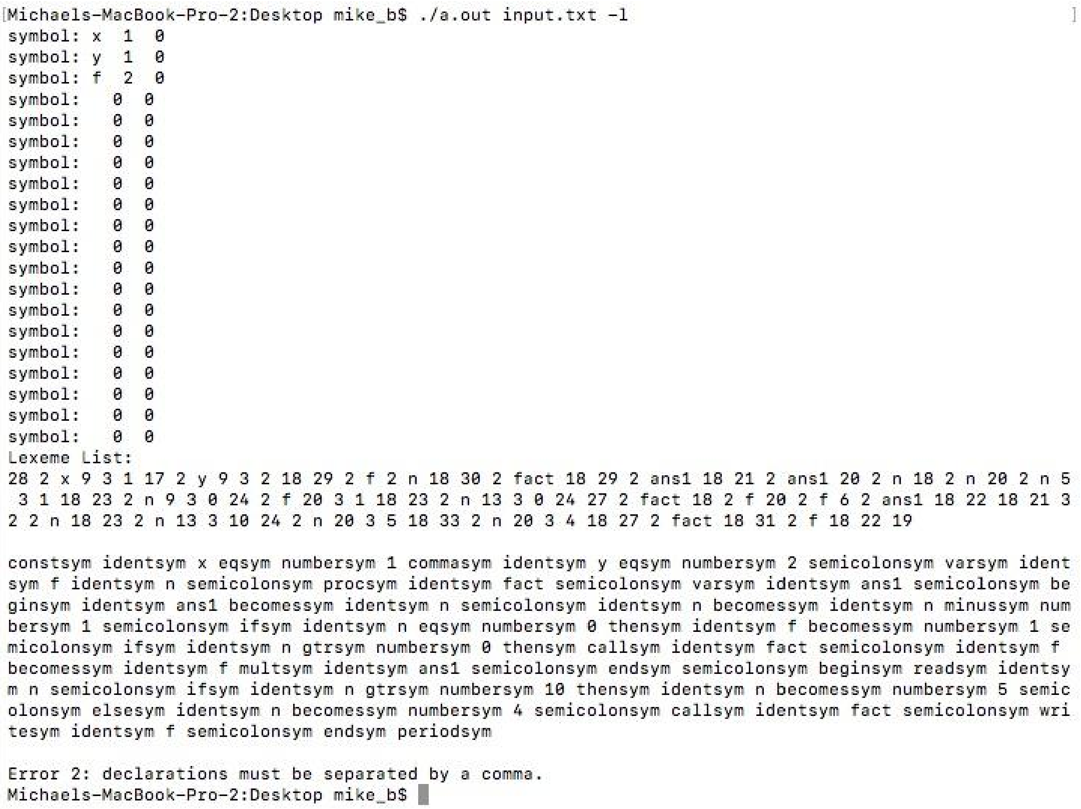
```
const x = 1, y = 2;
var , n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



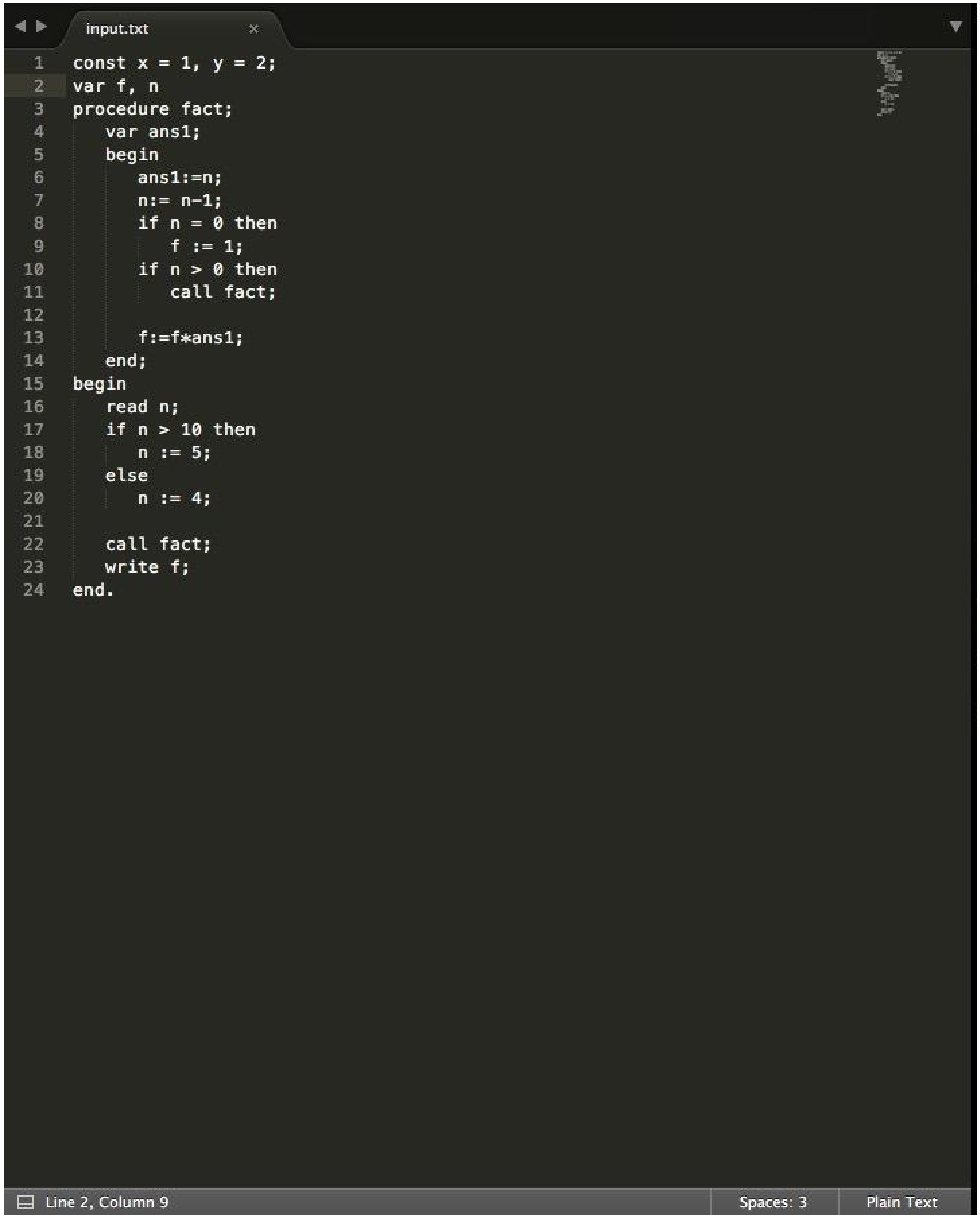
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n 5
3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 32
 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym comma
sym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym begi
nsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym numbe
rsym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 semi
colonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f be
comessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsym
n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semicol
onsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym write
sym identsym f semicolonsym endsym periodsym

Error 1: var, const, or procedure must be followed by an identifier.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



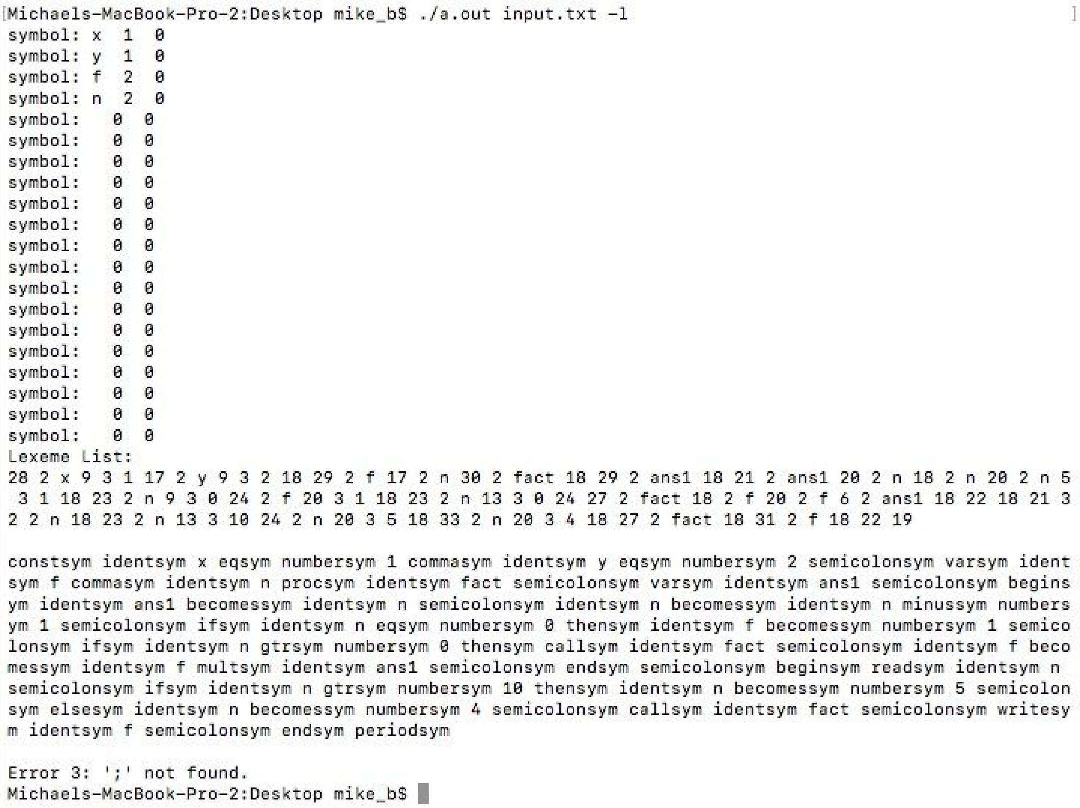
```
const x = 1, y = 2;
var f n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



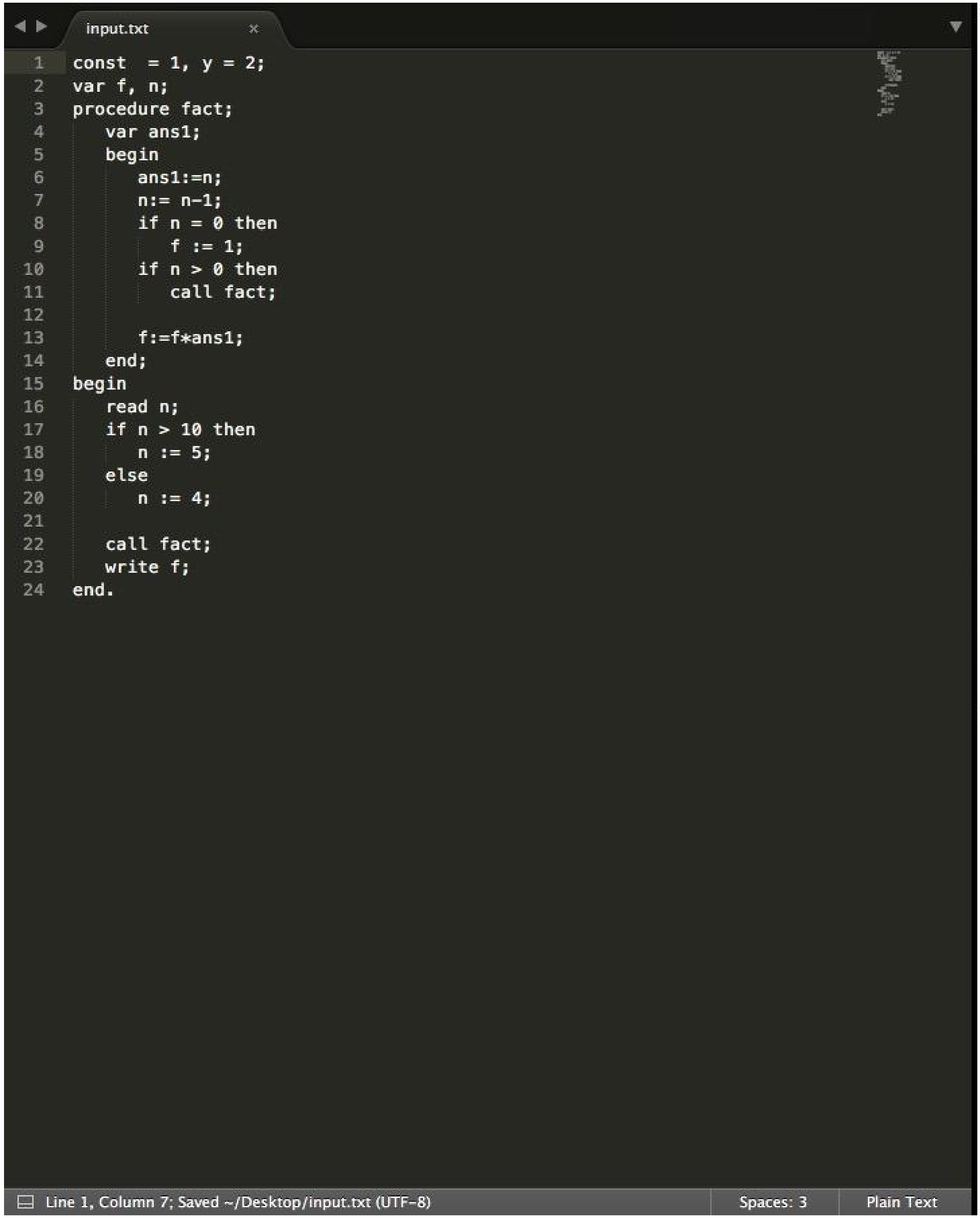
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n 5
 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 3
2 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym be
ginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym num
bersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 se
micolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f
becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsy
m n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semic
olonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym wri
tesym identsym f semicolonsym endsym periodsym

Error 2: declarations must be separated by a comma.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



```
const x = 1, y = 2;
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
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```

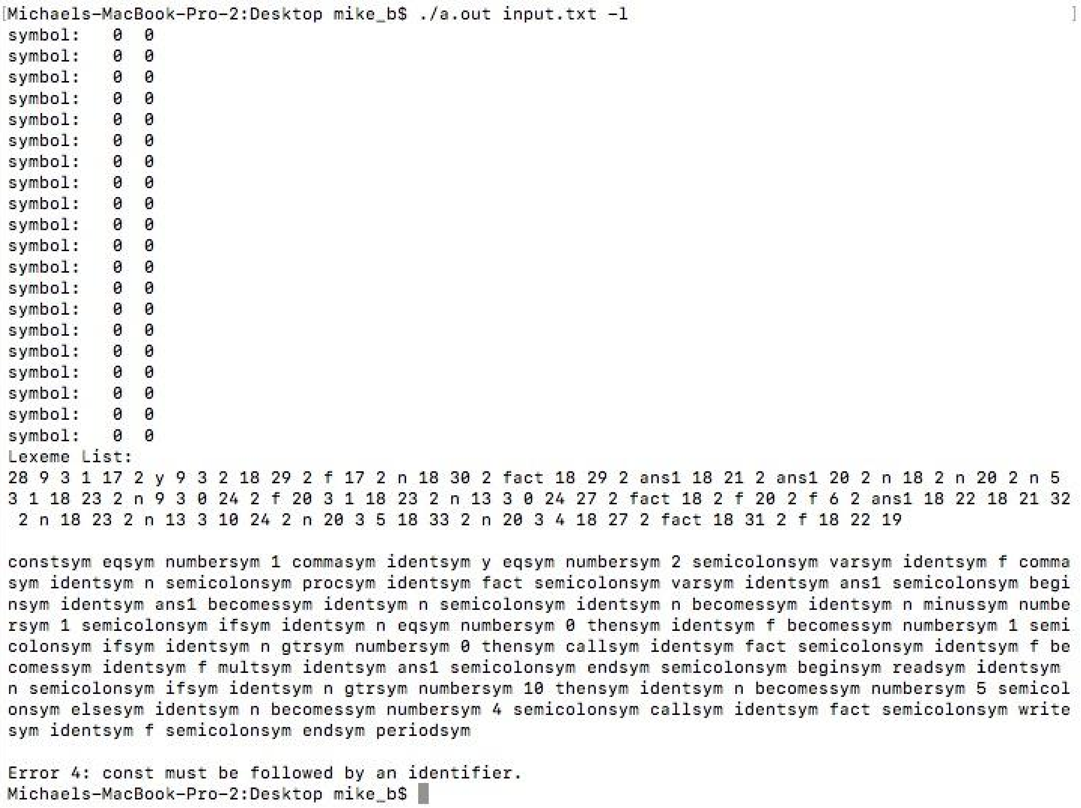


```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n 5
 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 3
2 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym begins
ym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym numbers
ym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 semico
lonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f beco
messym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsym n
semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semicolon
sym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym writesy
m identsym f semicolonsym endsym periodsym

Error 3: ';' not found.
Michaels-MacBook-Pro-2:Desktop mike_b$
```

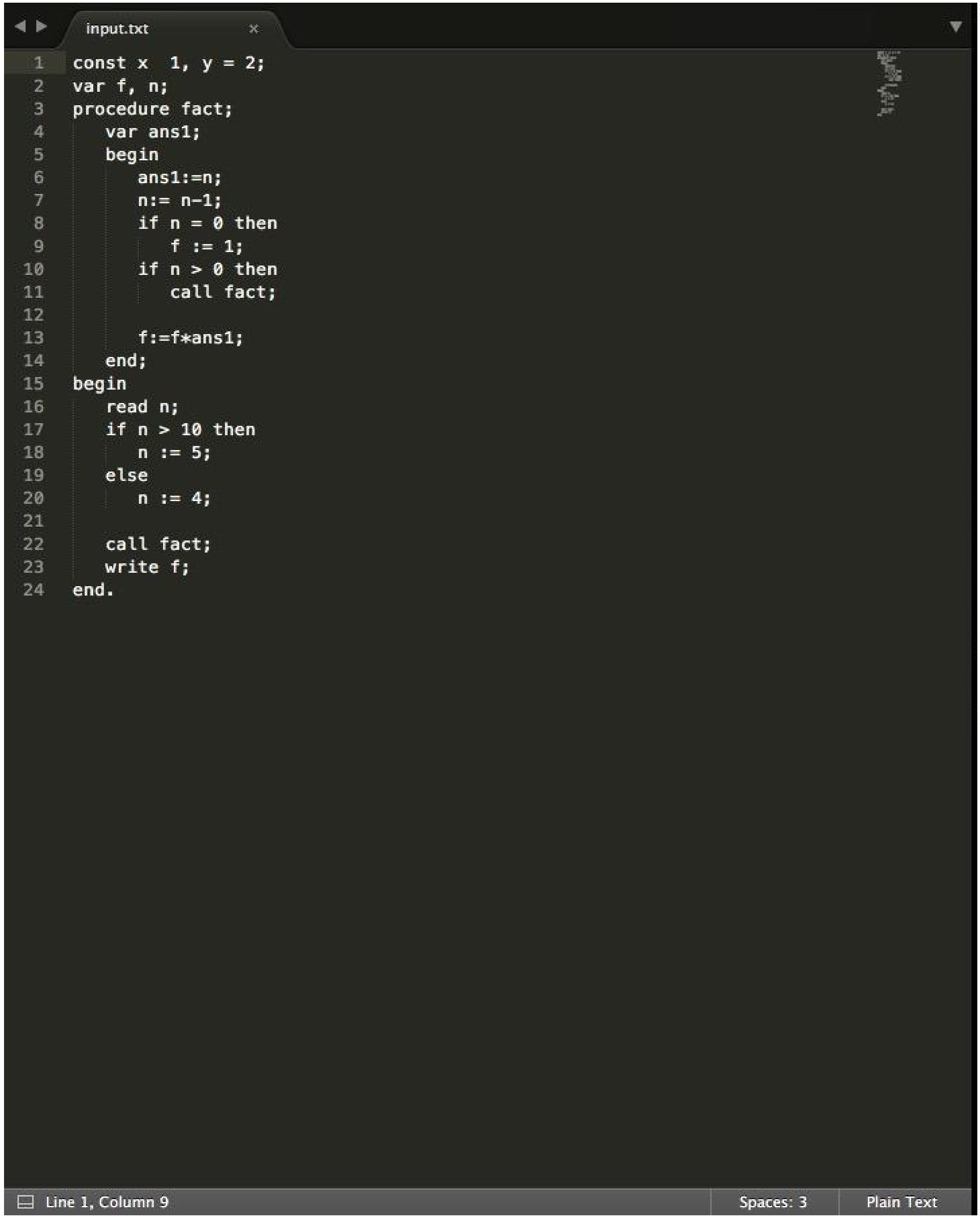


```
const  = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```

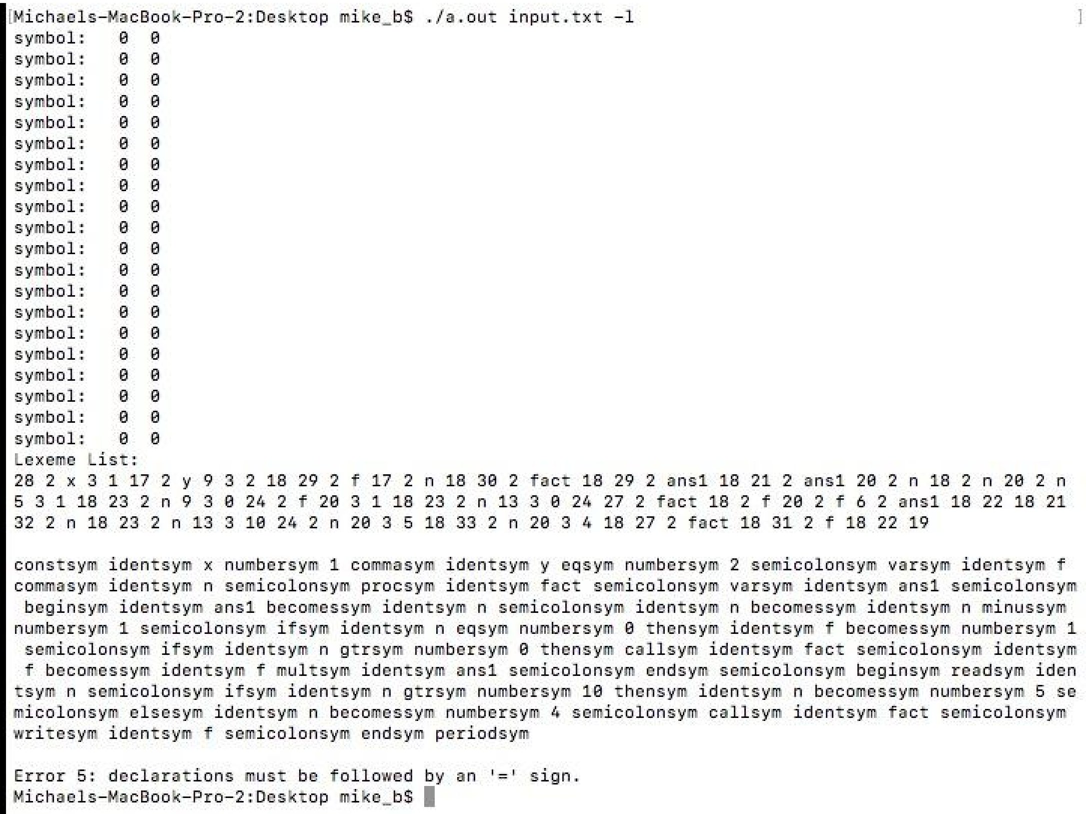
Line 1, Column 7; Saved ~/Desktop/input.txt (UTF-8) Spaces: 3 Plain Text



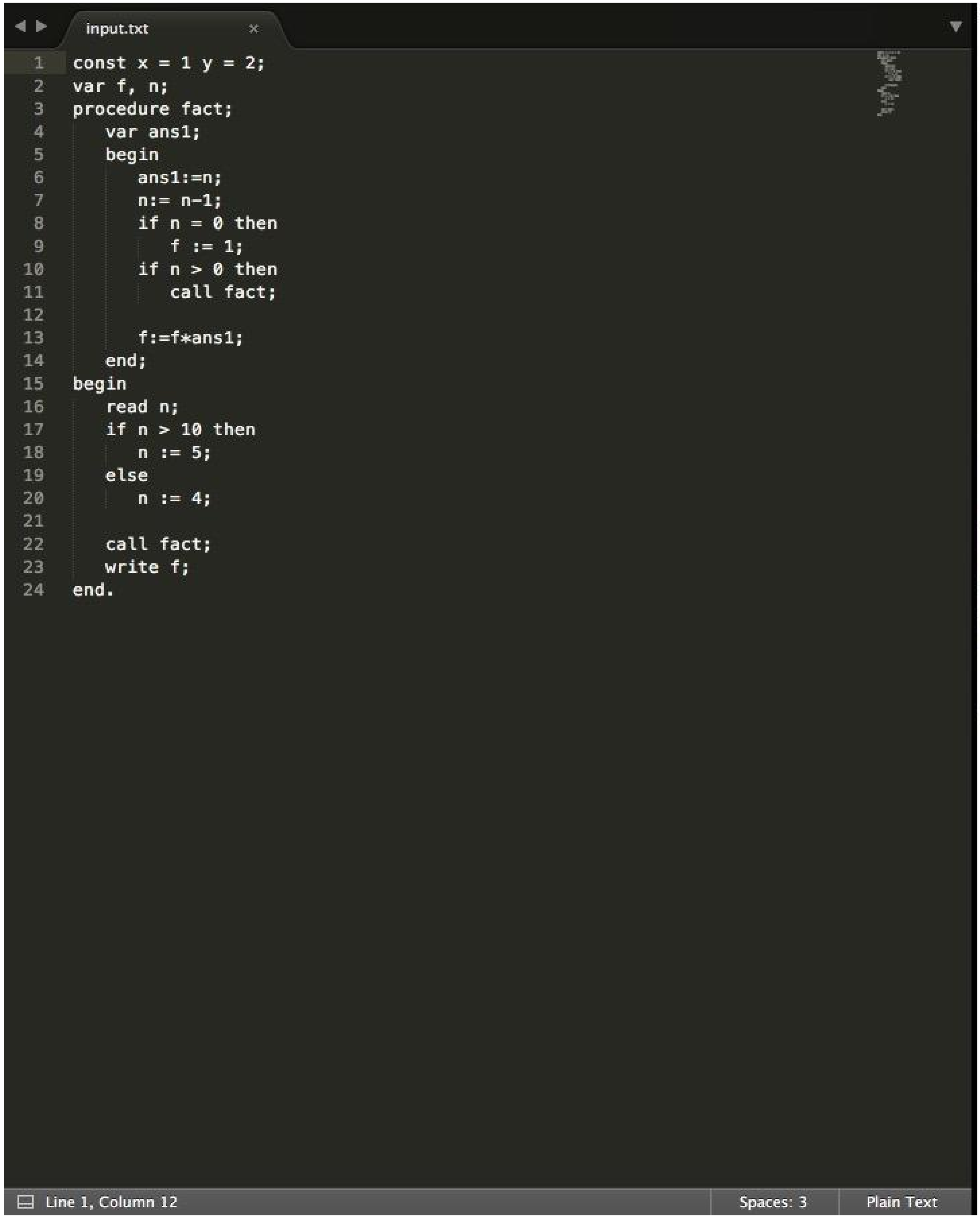
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n 5
3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 32
 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym identsym f comma
sym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym begi
nsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym numbe
rsym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 semi
colonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f be
comessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsym
n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semicol
onsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym write
sym identsym f semicolonsym endsym periodsym

Error 4: const must be followed by an identifier.
Michaels-MacBook-Pro-2:Desktop mike_b$ 
```



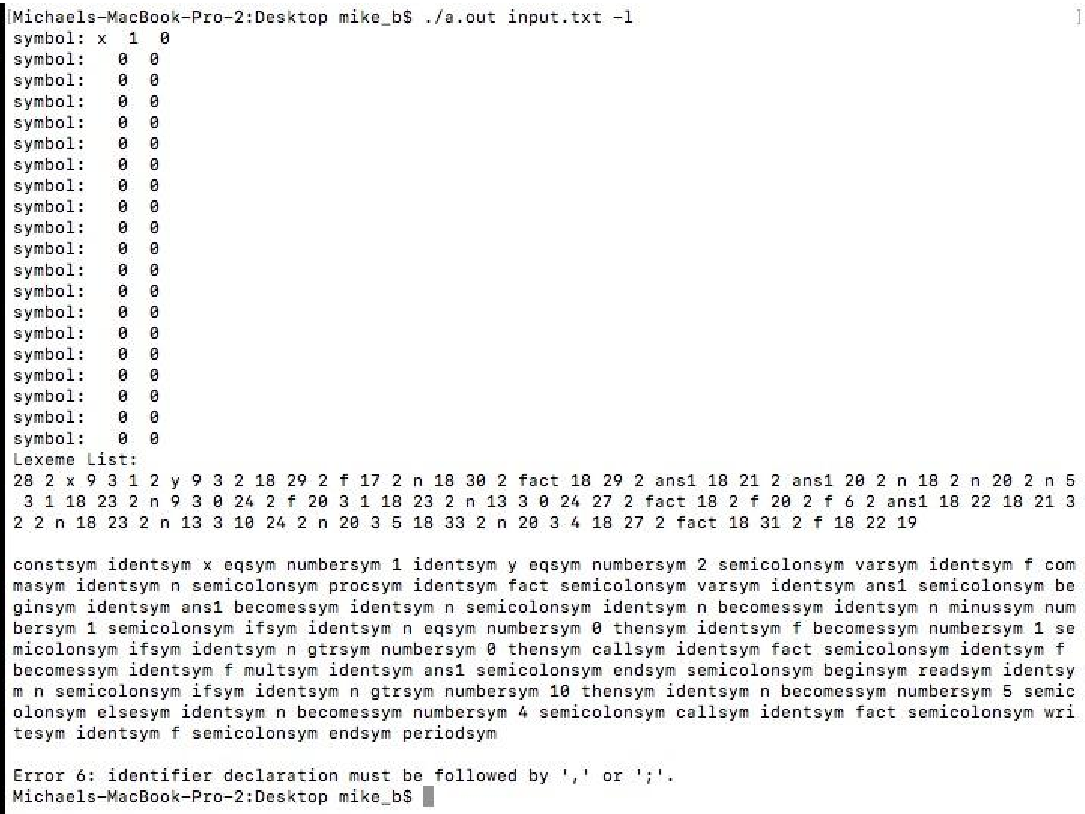
```
const x  1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



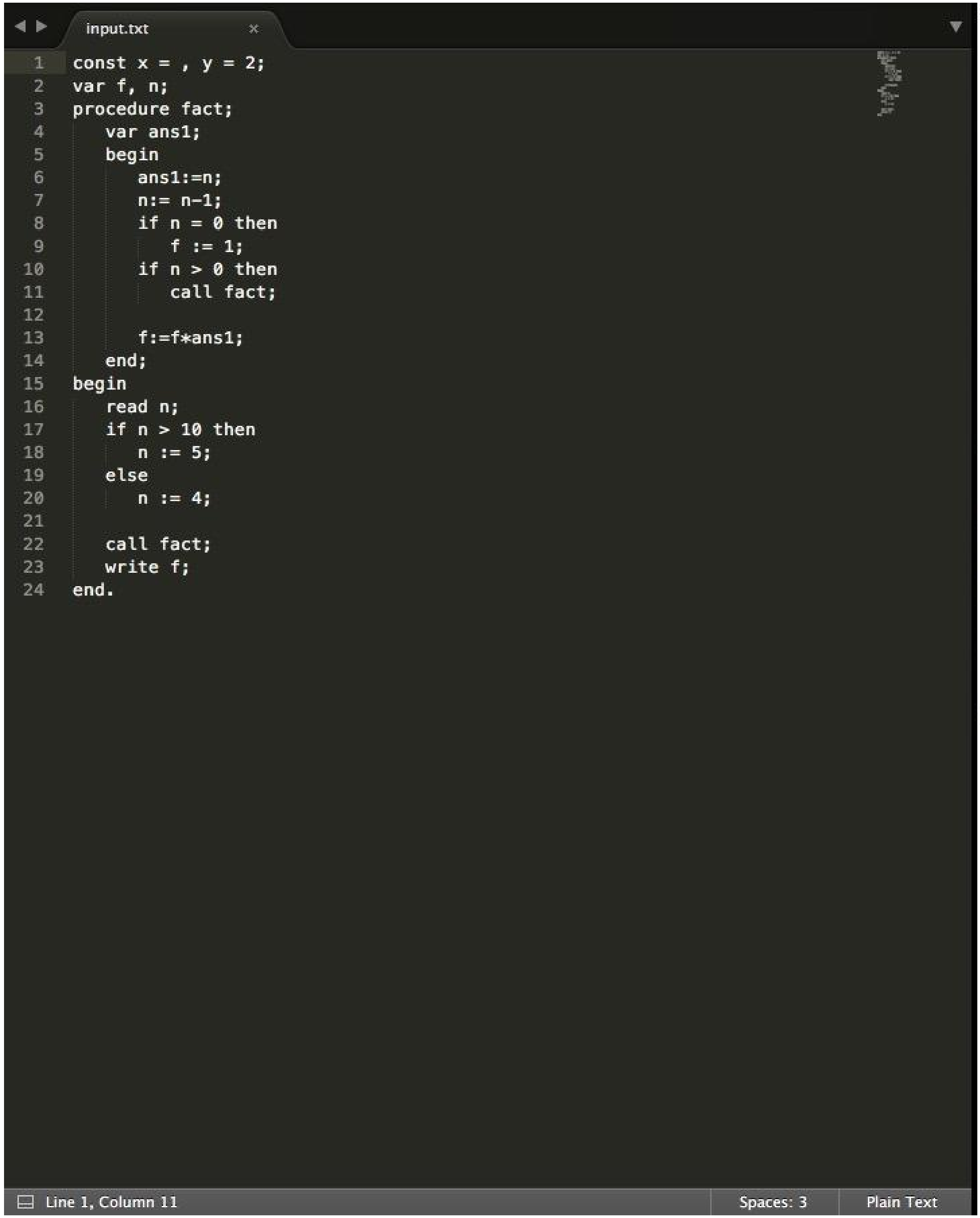
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n
5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21
32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym identsym f
commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym
 beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym
numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1
 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym
 f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym iden
tsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 se
micolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym
writesym identsym f semicolonsym endsym periodsym

Error 5: declarations must be followed by an '=' sign.
Michaels-MacBook-Pro-2:Desktop mike_b$ 
```



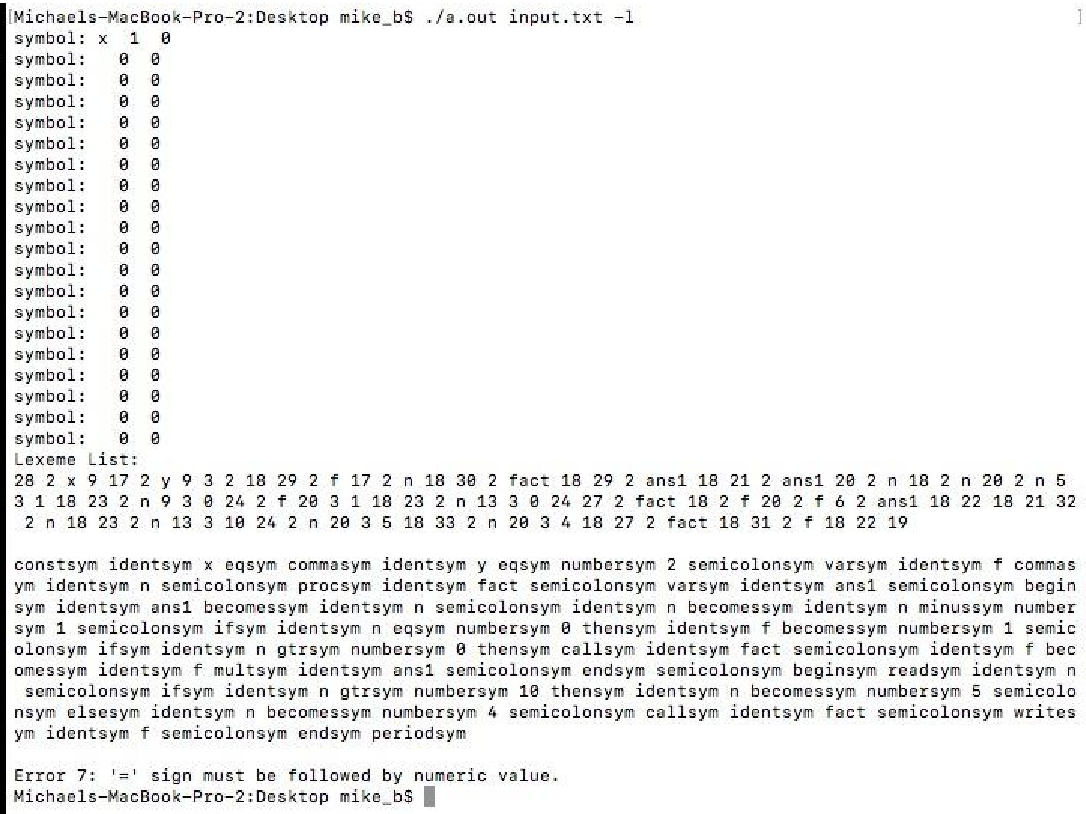
```
const x = 1 y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



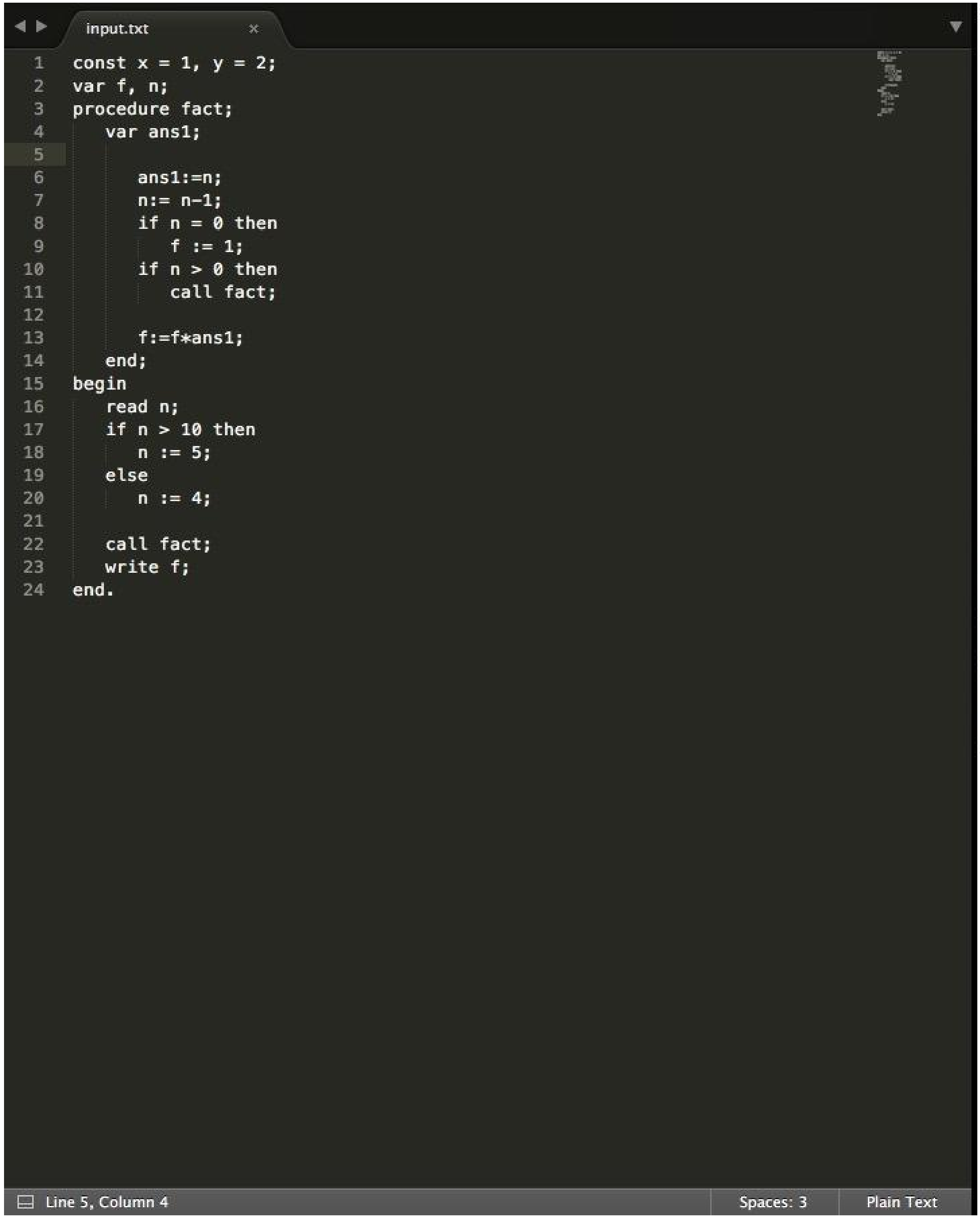
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n 5
 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 3
2 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 identsym y eqsym numbersym 2 semicolonsym varsym identsym f com
masym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym be
ginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym num
bersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 se
micolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f
becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsy
m n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semic
olonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym wri
tesym identsym f semicolonsym endsym periodsym

Error 6: identifier declaration must be followed by ',' or ';'.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



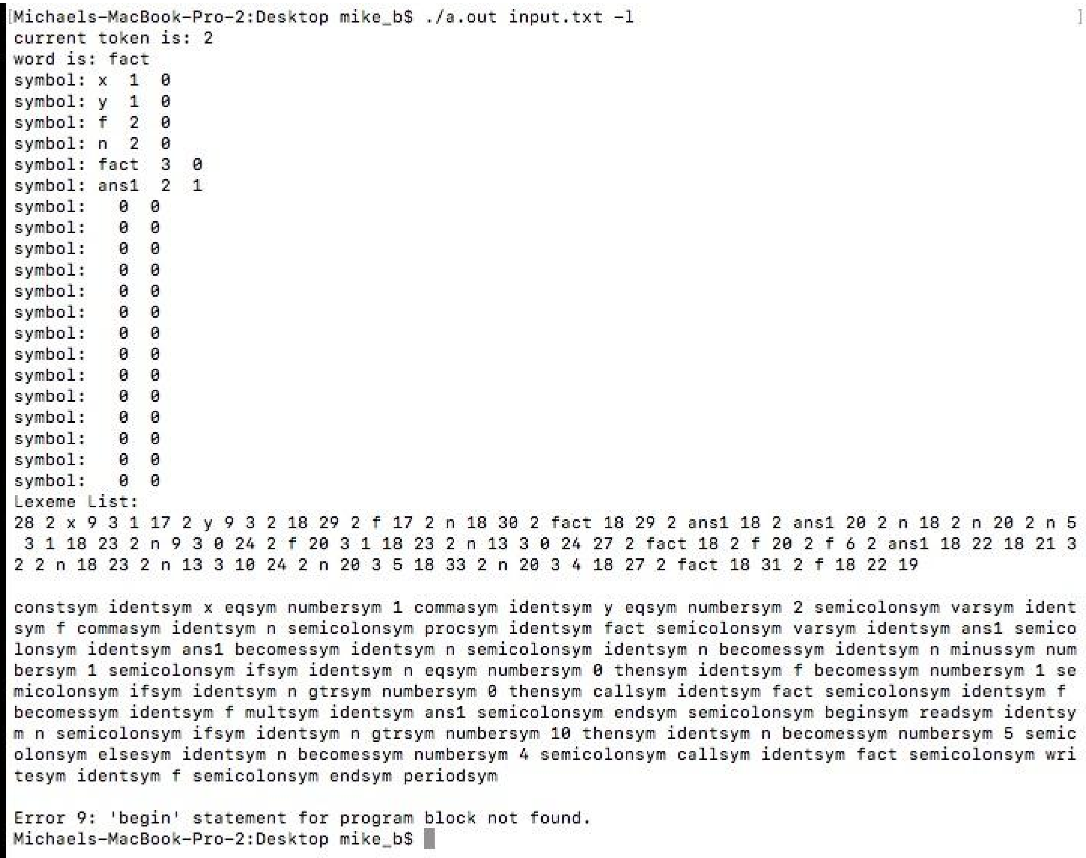
```
const x = , y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



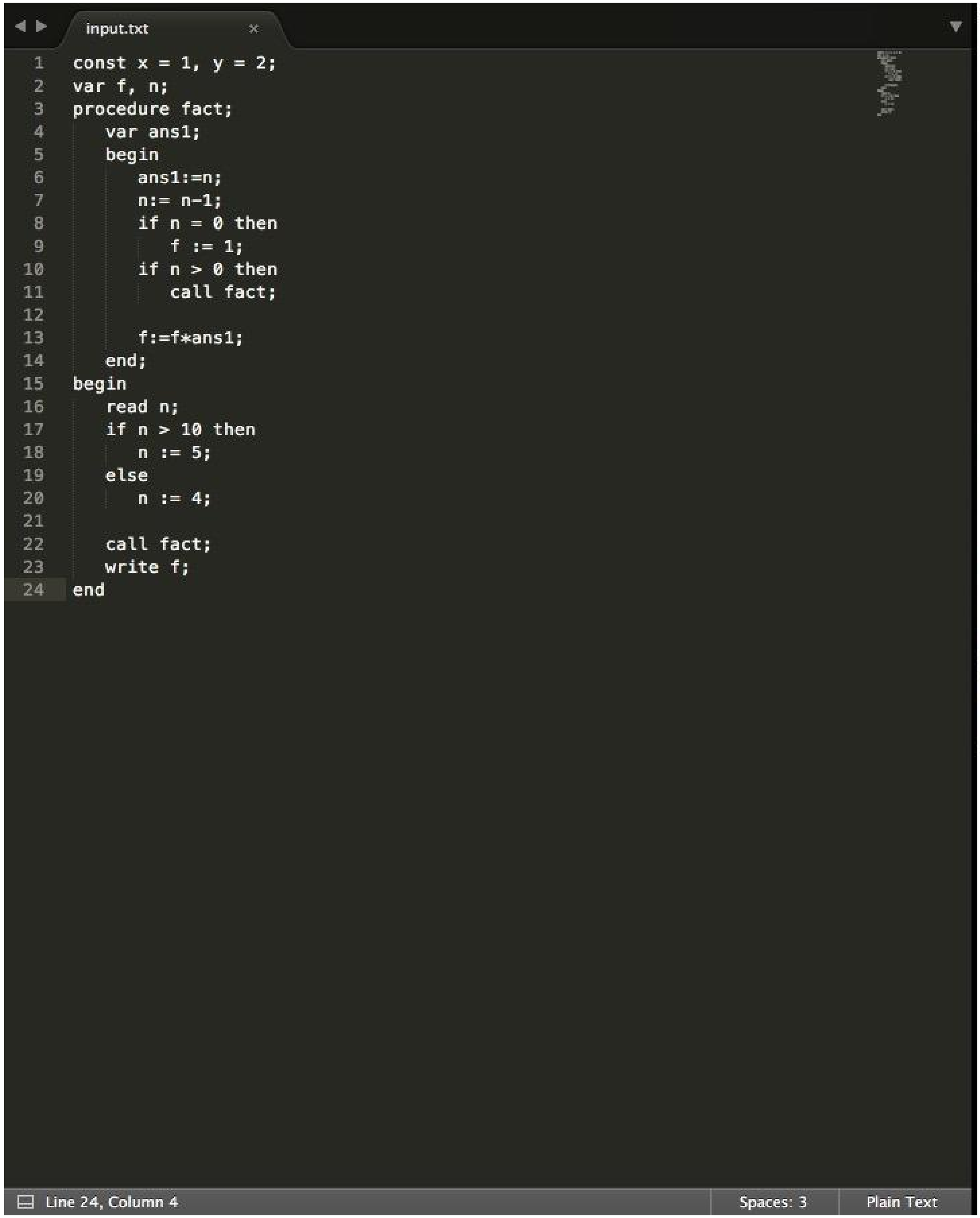
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2 n 5
3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 32
 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym commasym identsym y eqsym numbersym 2 semicolonsym varsym identsym f commas
ym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semicolonsym begin
sym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym number
sym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 semic
olonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f bec
omessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsym n
 semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semicolo
nsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym writes
ym identsym f semicolonsym endsym periodsym

Error 7: '=' sign must be followed by numeric value.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



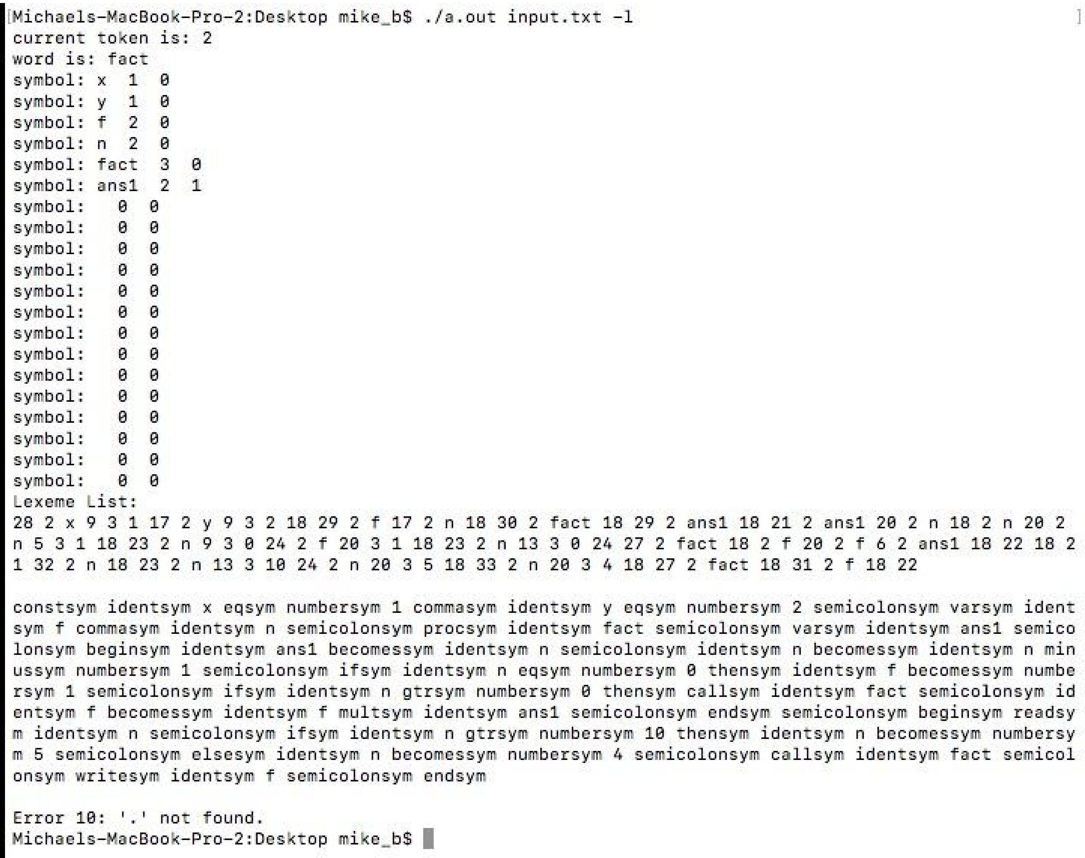
```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



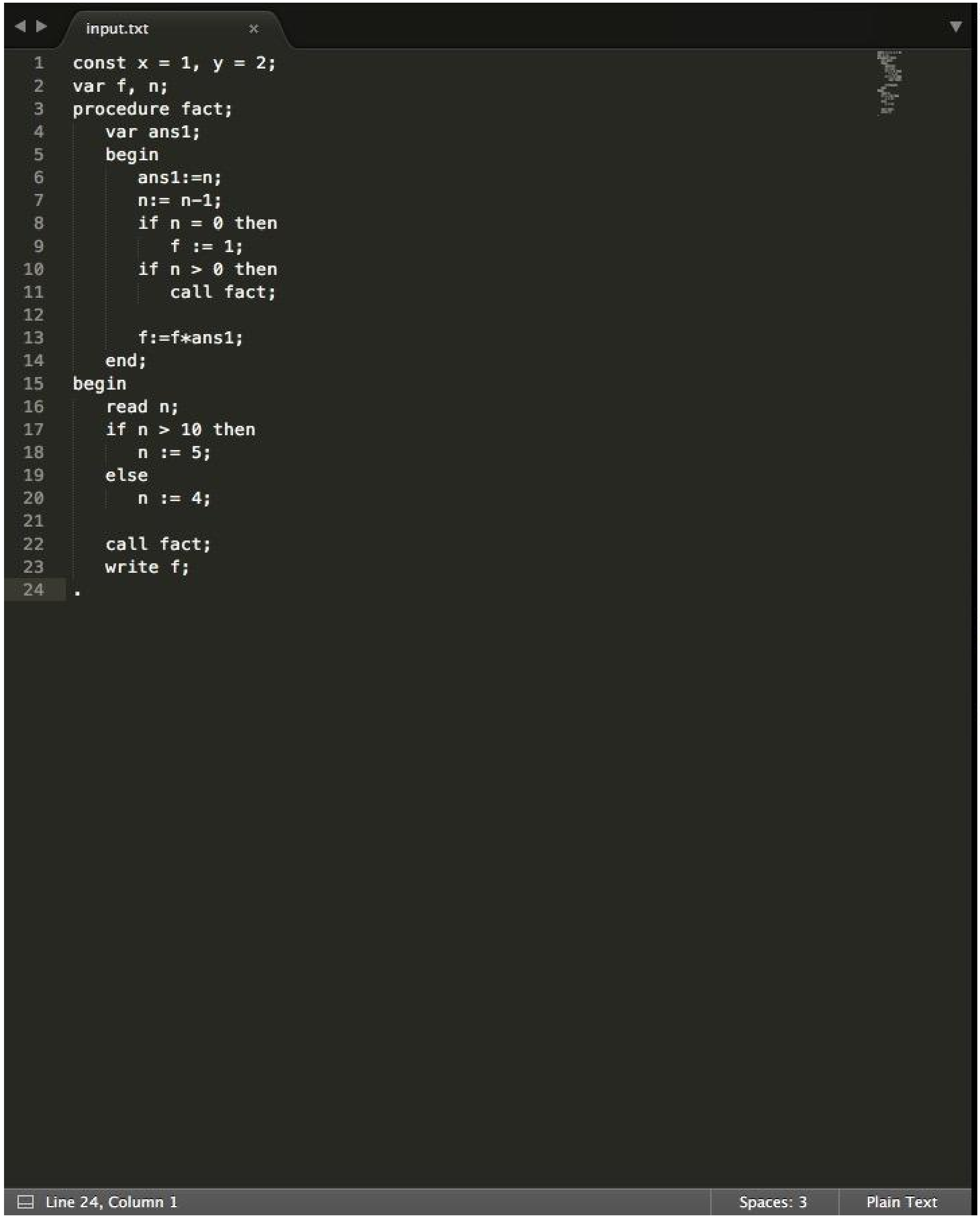
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
current token is: 2
word is: fact
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 2 ans1 20 2 n 18 2 n 20 2 n 5
 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 3
2 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n minussym num
bersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 se
micolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f
becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsy
m n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semic
olonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym wri
tesym identsym f semicolonsym endsym periodsym

Error 9: 'begin' statement for program block not found.
Michaels-MacBook-Pro-2:Desktop mike_b$ 
```



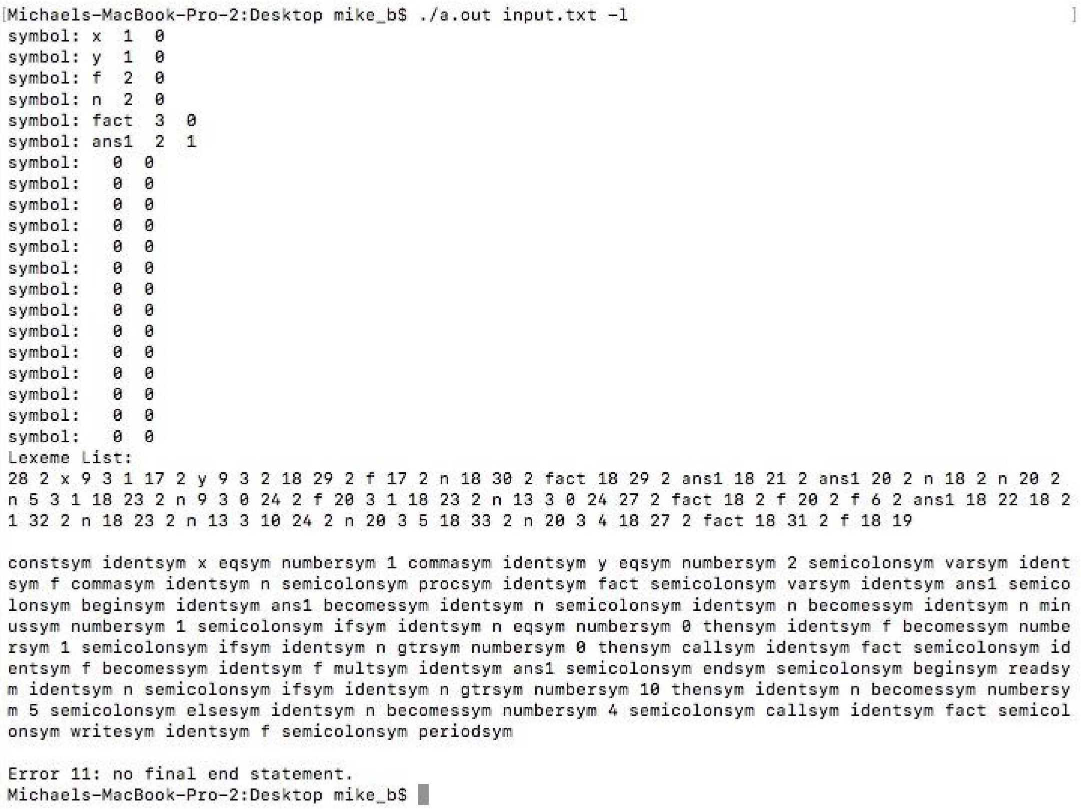
```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end
```



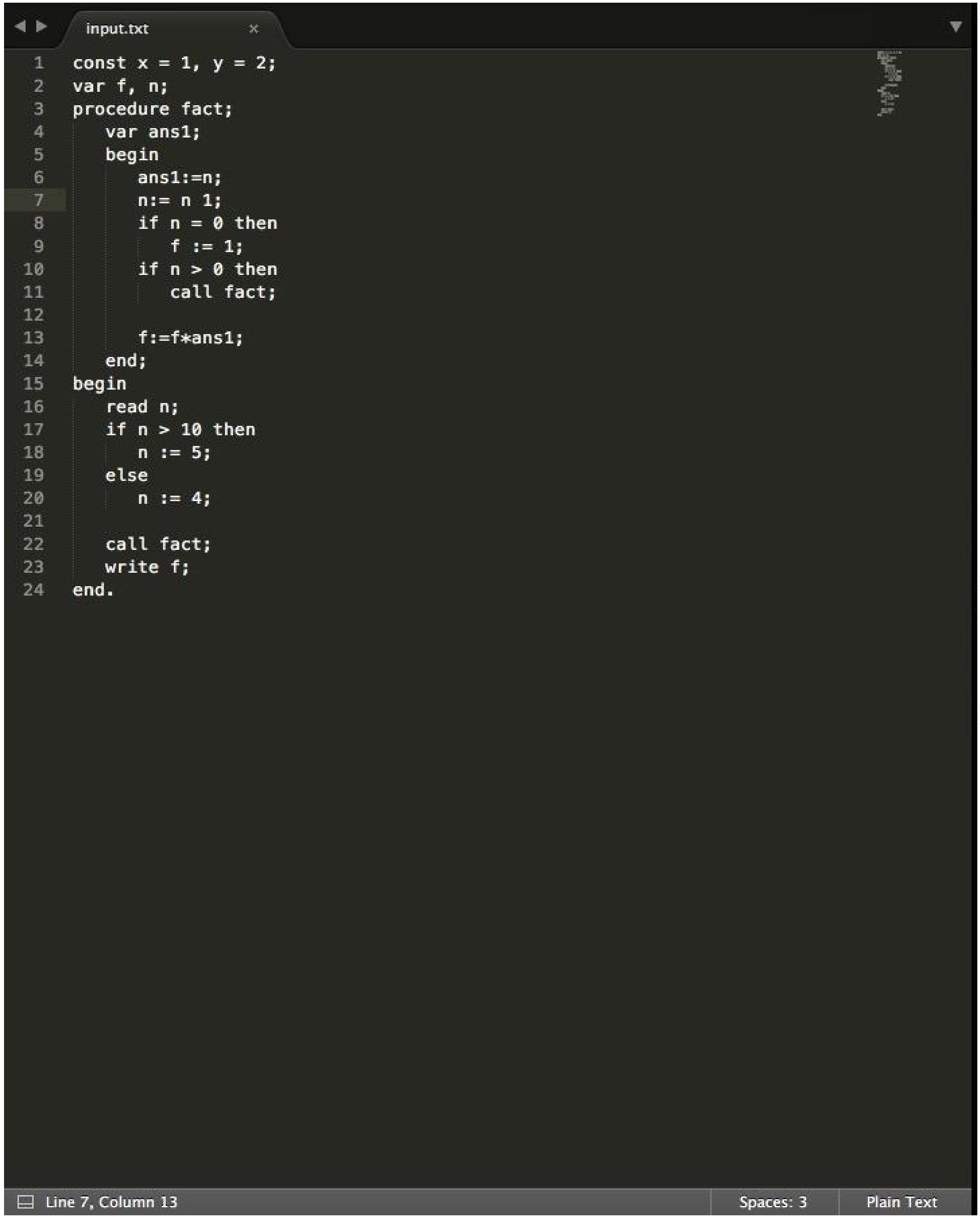
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
current token is: 2
word is: fact
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 2
1 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n min
ussym numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbe
rsym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym id
entsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsy
m identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersy
m 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicol
onsym writesym identsym f semicolonsym endsym

Error 10: '.' not found.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



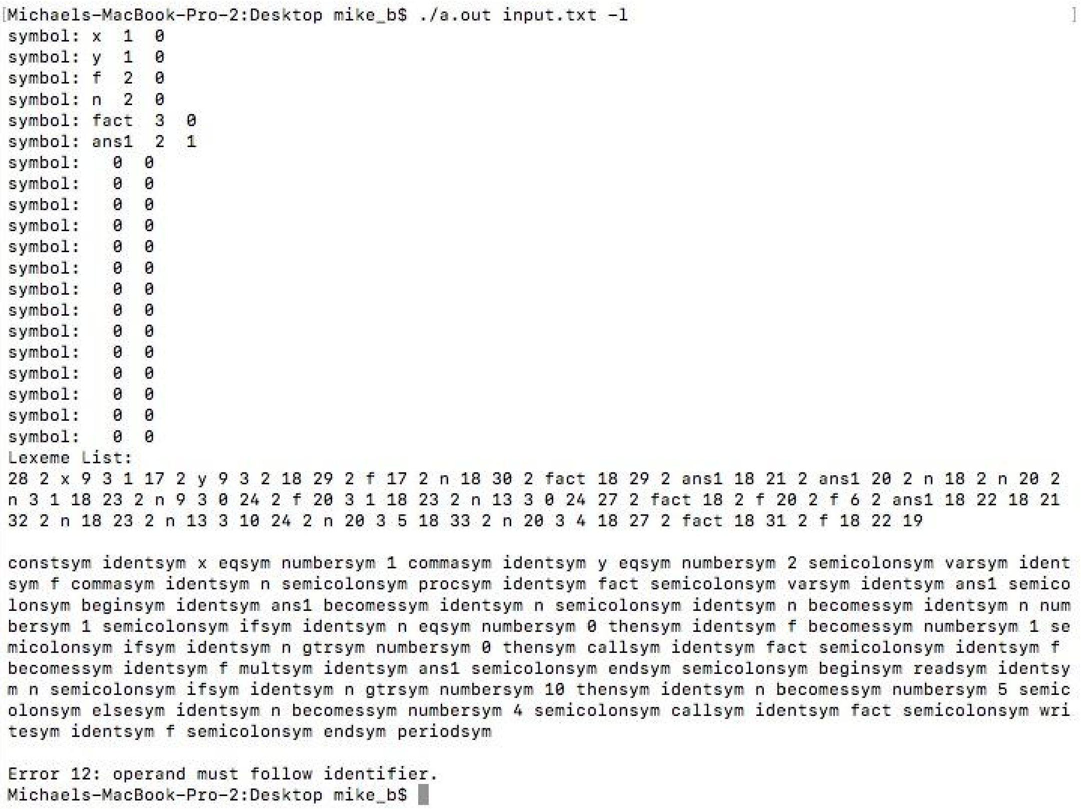
```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
.
```



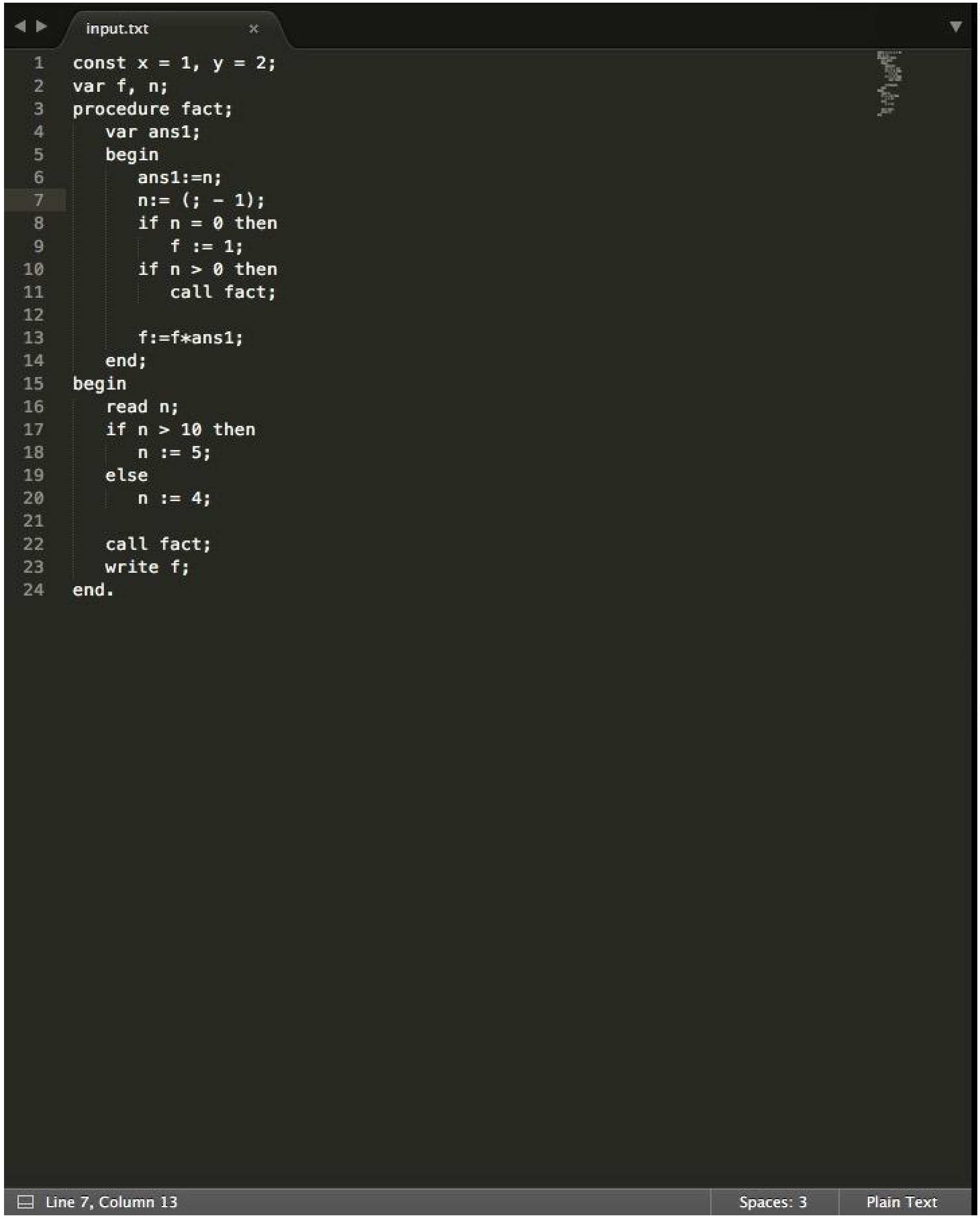
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 2
1 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n min
ussym numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbe
rsym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym id
entsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsy
m identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersy
m 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicol
onsym writesym identsym f semicolonsym periodsym

Error 11: no final end statement.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



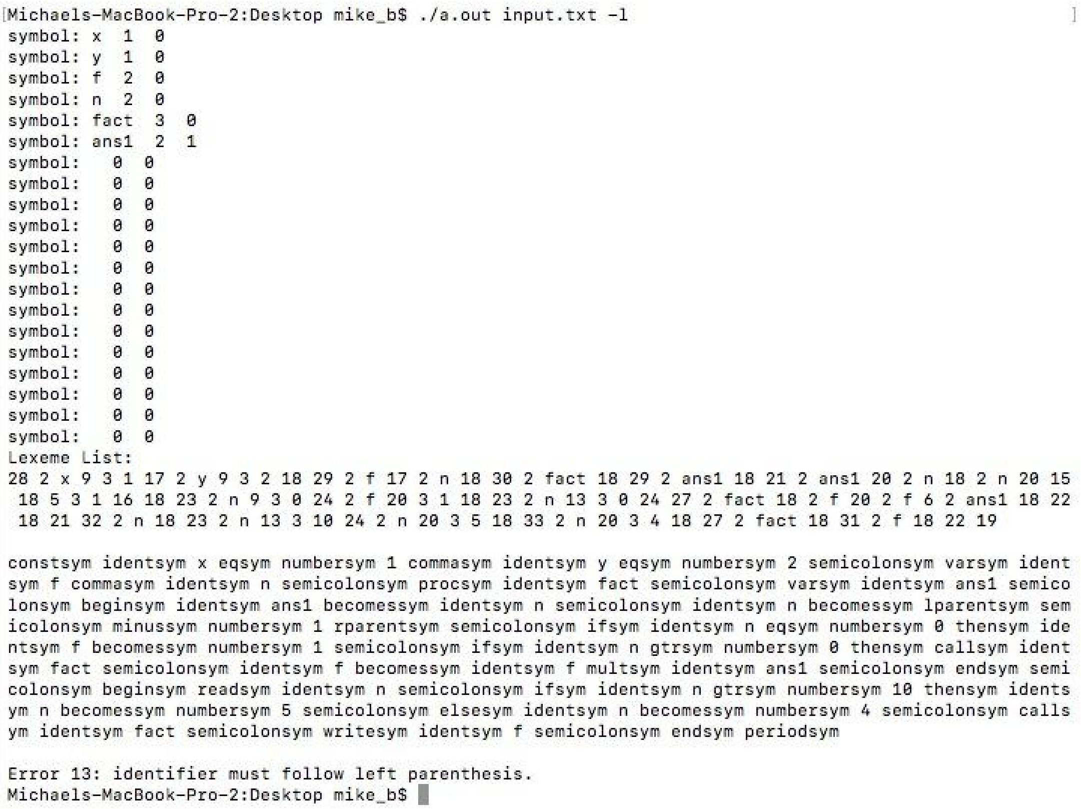
```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n 1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



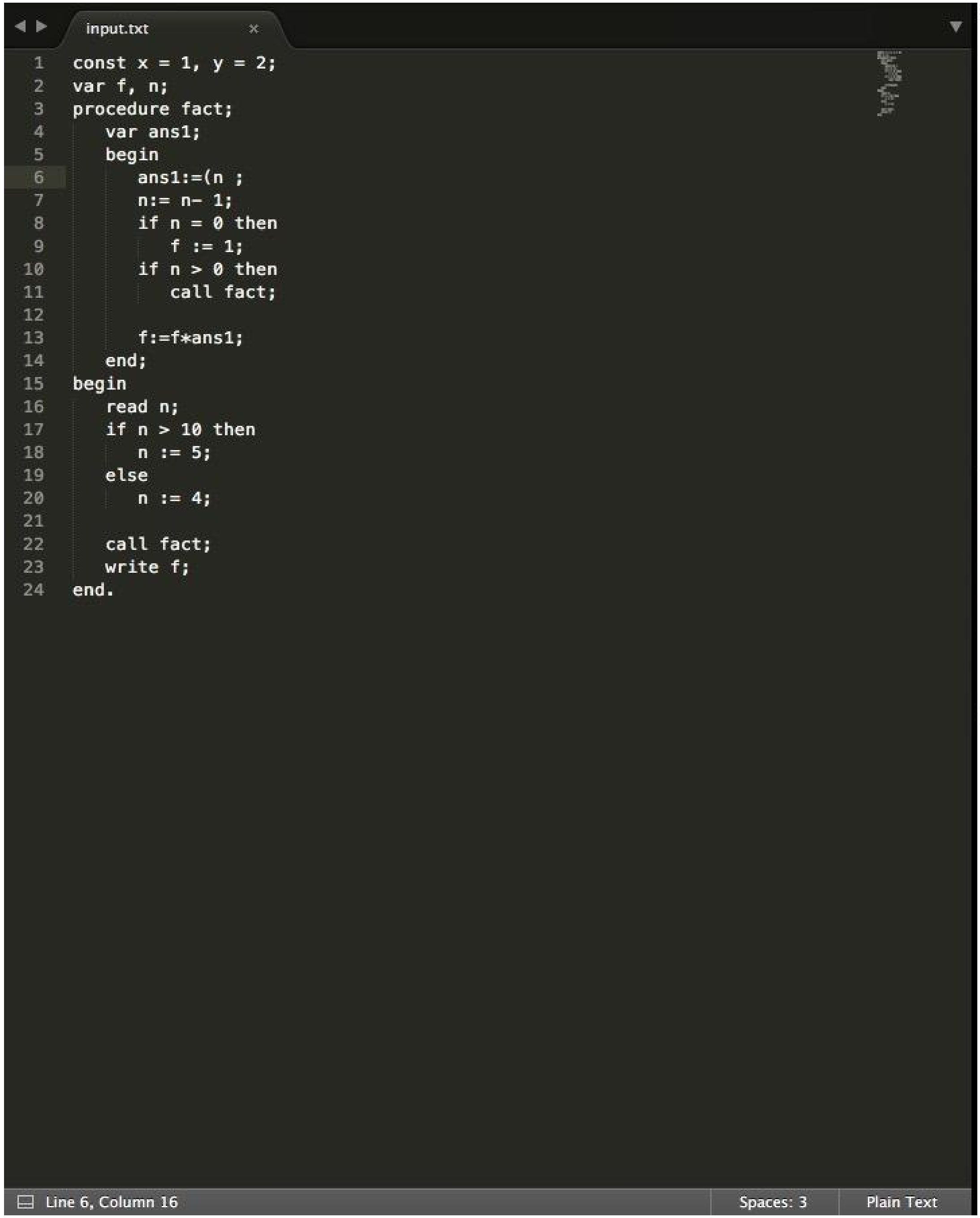
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21
32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n num
bersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym 1 se
micolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f
becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym identsy
m n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semic
olonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym wri
tesym identsym f semicolonsym endsym periodsym

Error 12: operand must follow identifier.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= (; - 1);
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```

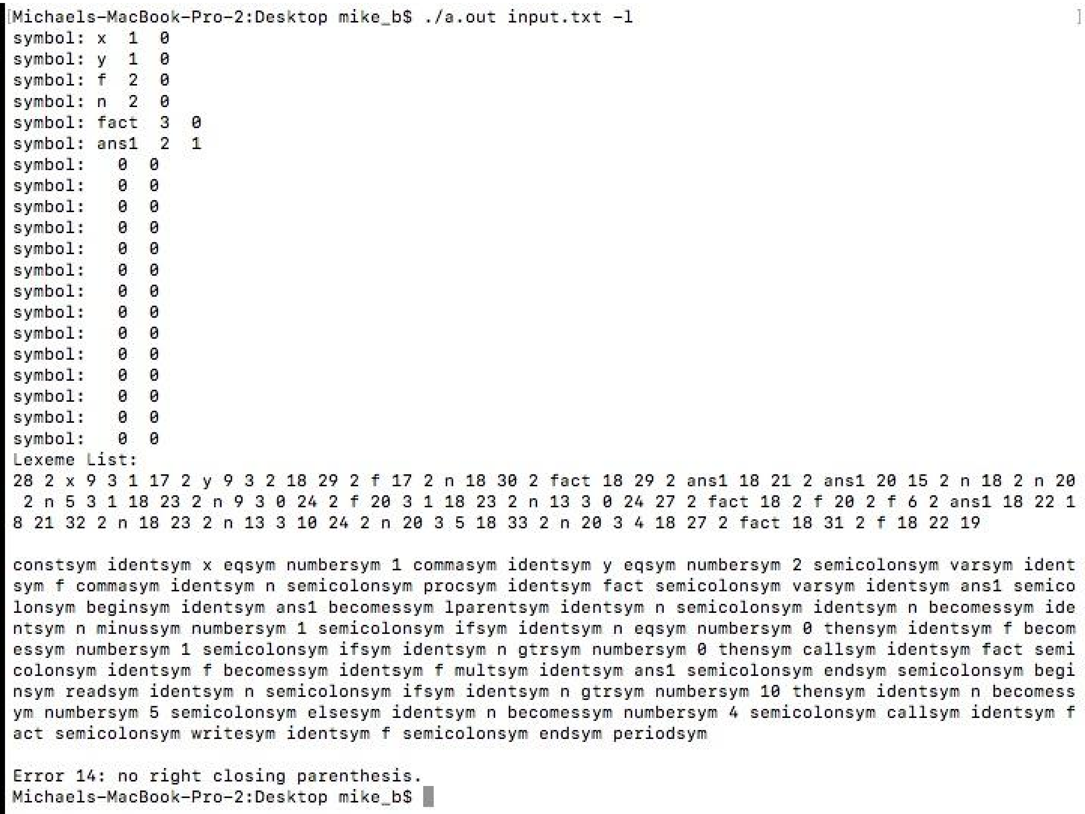


```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 15
 18 5 3 1 16 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22
 18 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym lparentsym sem
icolonsym minussym numbersym 1 rparentsym semicolonsym ifsym identsym n eqsym numbersym 0 thensym ide
ntsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym ident
sym fact semicolonsym identsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semi
colonsym beginsym readsym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym idents
ym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym calls
ym identsym fact semicolonsym writesym identsym f semicolonsym endsym periodsym

Error 13: identifier must follow left parenthesis.
Michaels-MacBook-Pro-2:Desktop mike_b$ 
```

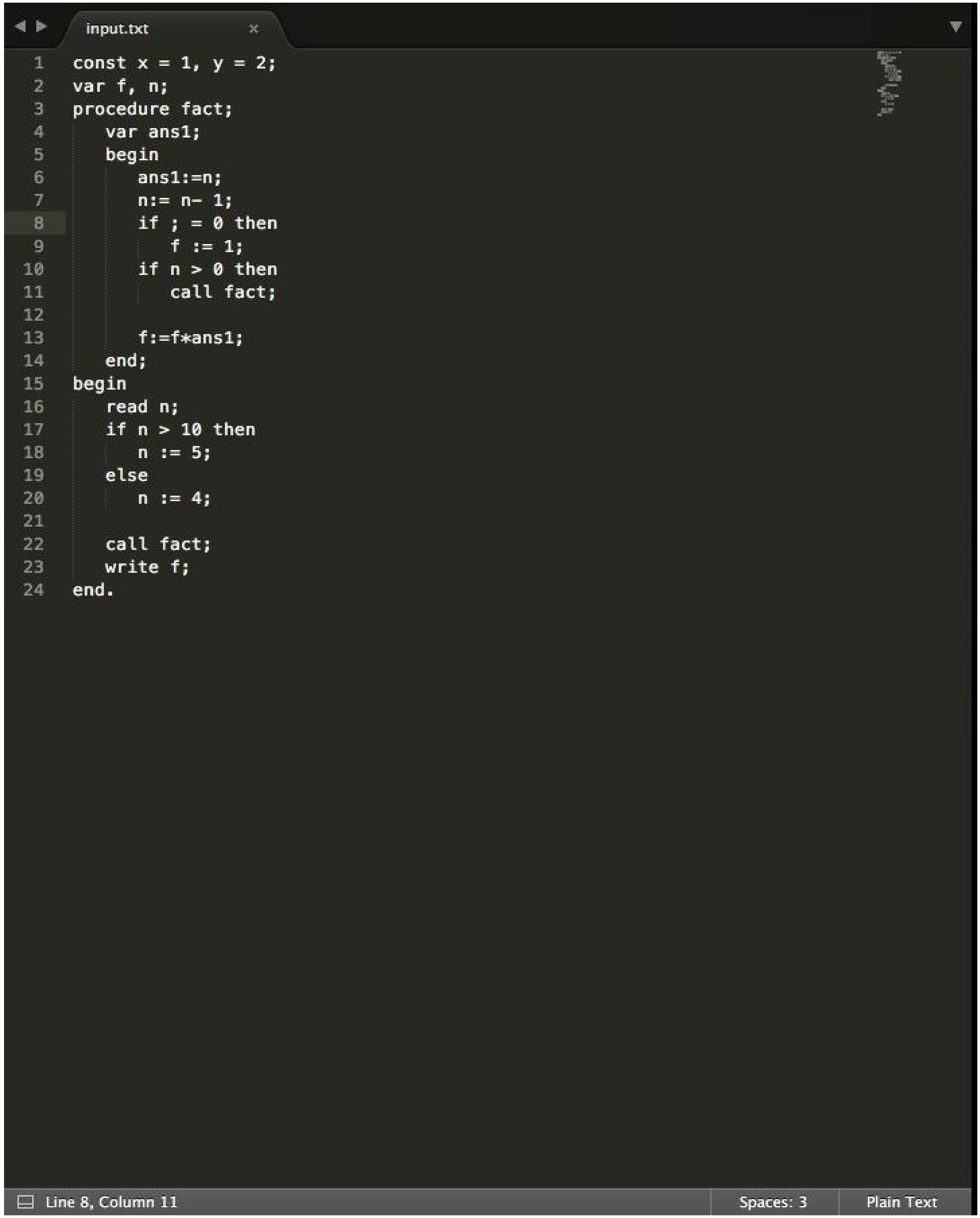


```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=(n ;
      n:= n- 1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```

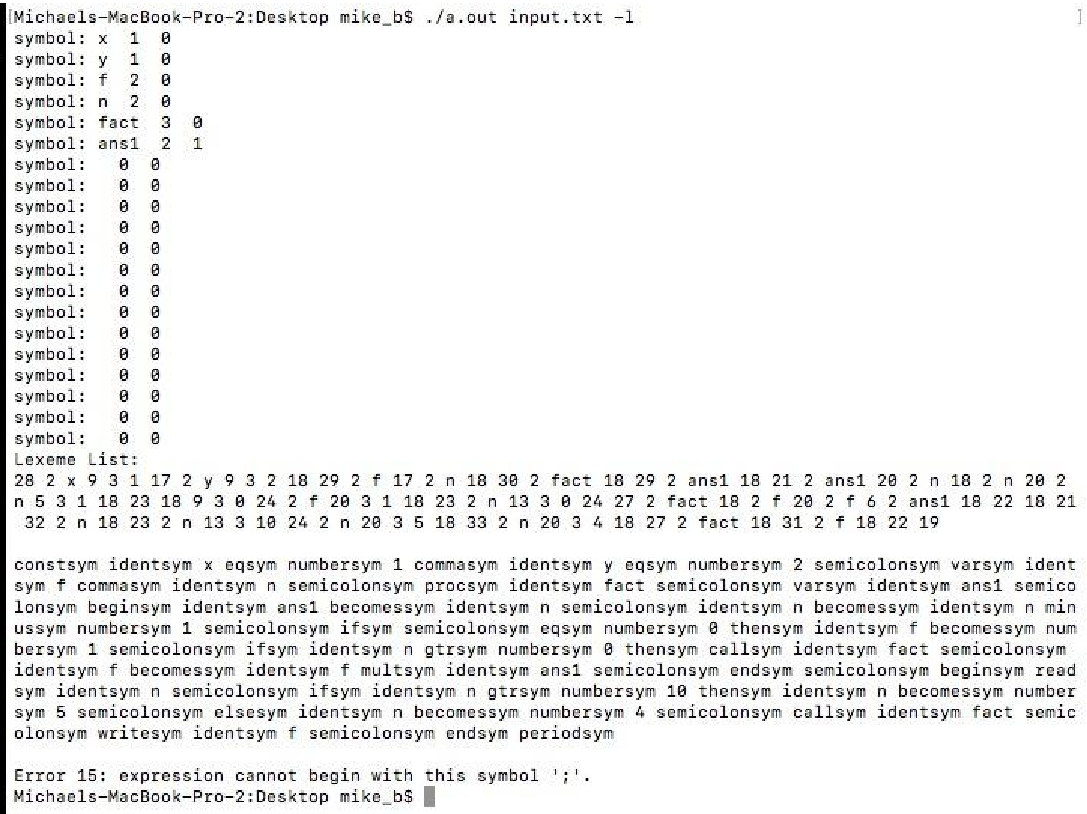
Line 6, Column 16 | Spaces: 3 | Plain Text



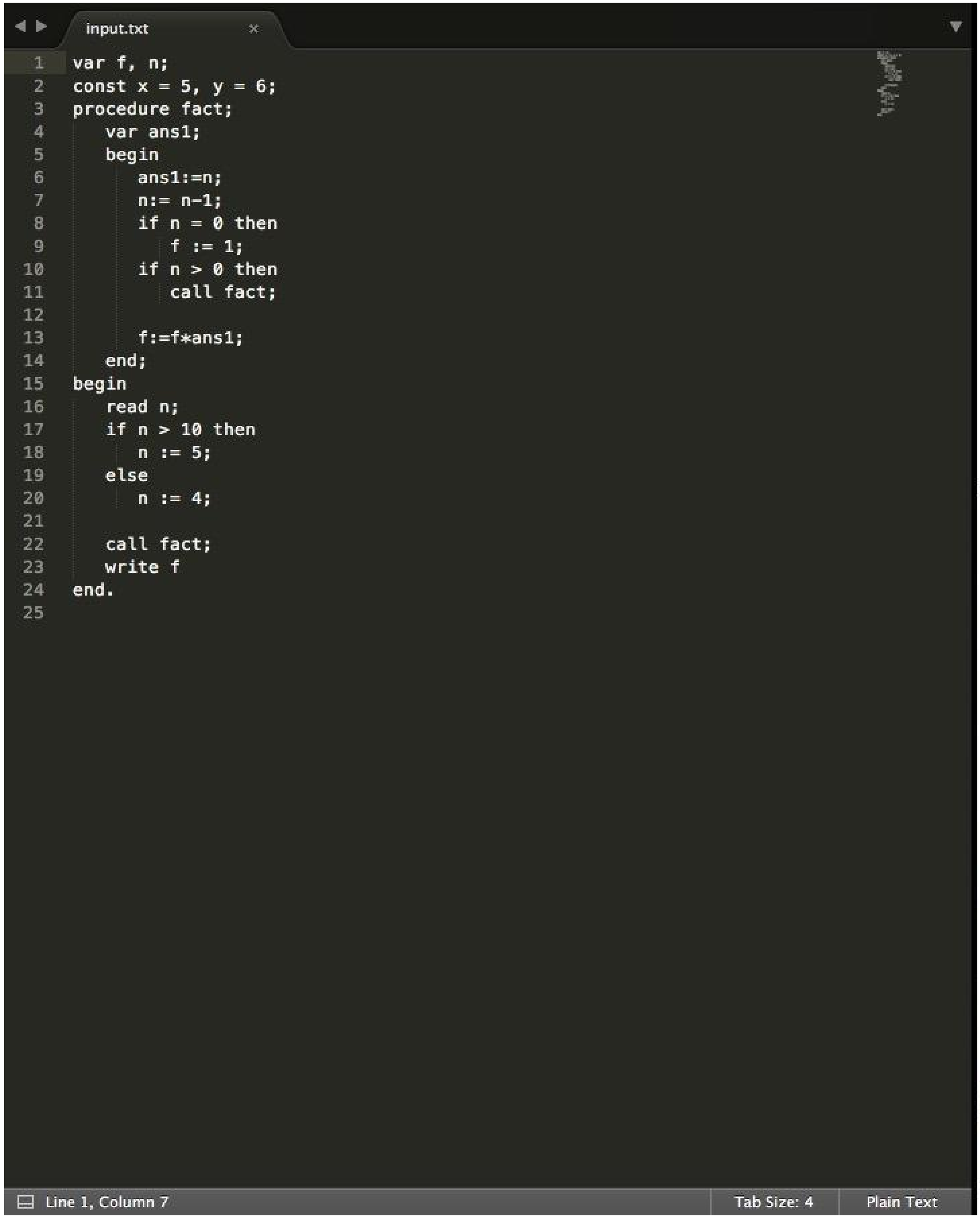
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 15 2 n 18 2 n 20
 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 1
8 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym lparentsym identsym n semicolonsym identsym n becomessym ide
ntsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becom
essym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semi
colonsym identsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym begi
nsym readsym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomess
ym numbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym f
act semicolonsym writesym identsym f semicolonsym endsym periodsym

Error 14: no right closing parenthesis.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



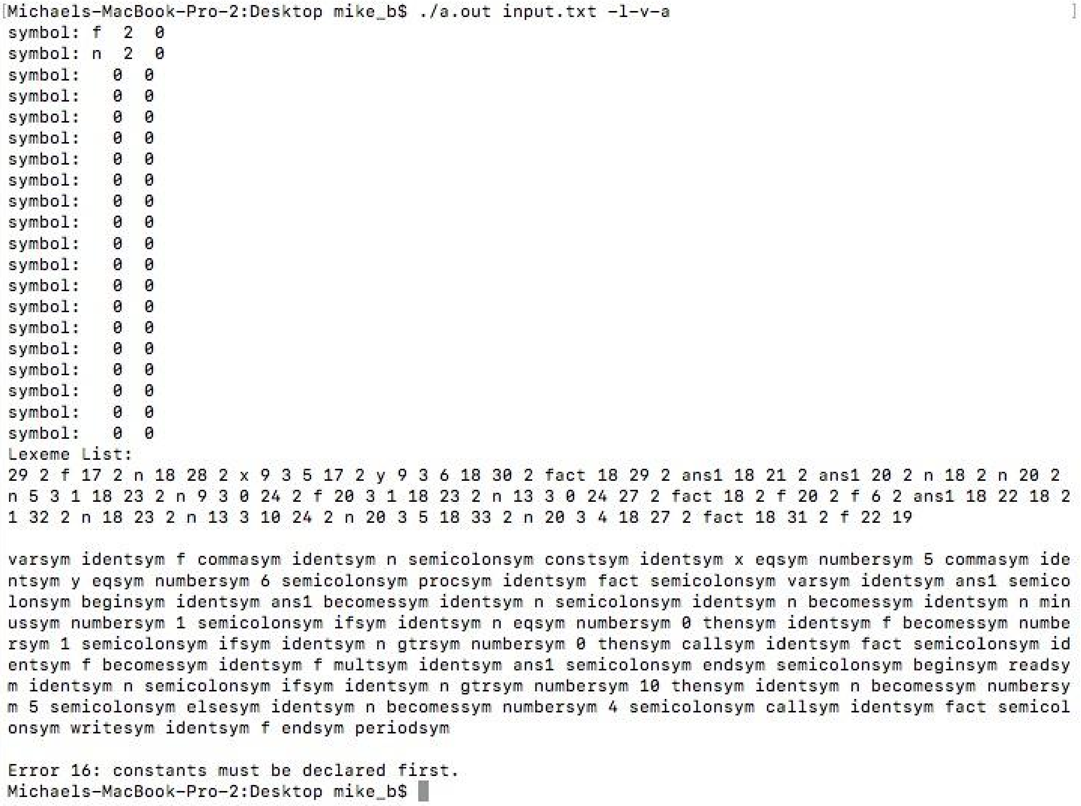
```
const x = 1, y = 2;
var f, n;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n- 1;
      if ; = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f;
end.
```



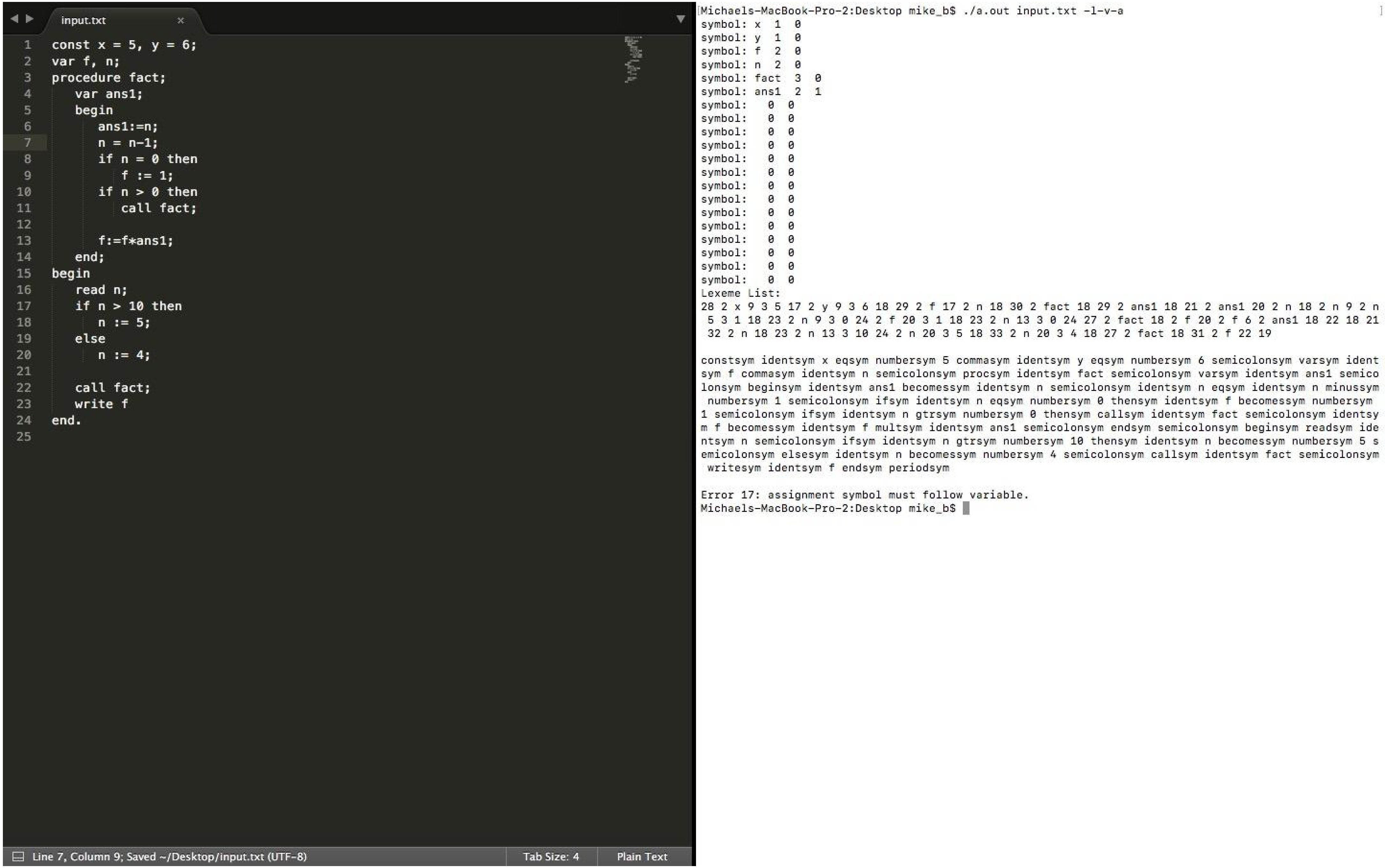
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 1 17 2 y 9 3 2 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 5 3 1 18 23 18 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21
 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 18 22 19

constsym identsym x eqsym numbersym 1 commasym identsym y eqsym numbersym 2 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n min
ussym numbersym 1 semicolonsym ifsym semicolonsym eqsym numbersym 0 thensym identsym f becomessym num
bersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym
identsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym read
sym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym number
sym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semic
olonsym writesym identsym f semicolonsym endsym periodsym

Error 15: expression cannot begin with this symbol ';'.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



```
var f, n;
const x = 5, y = 6;
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
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```

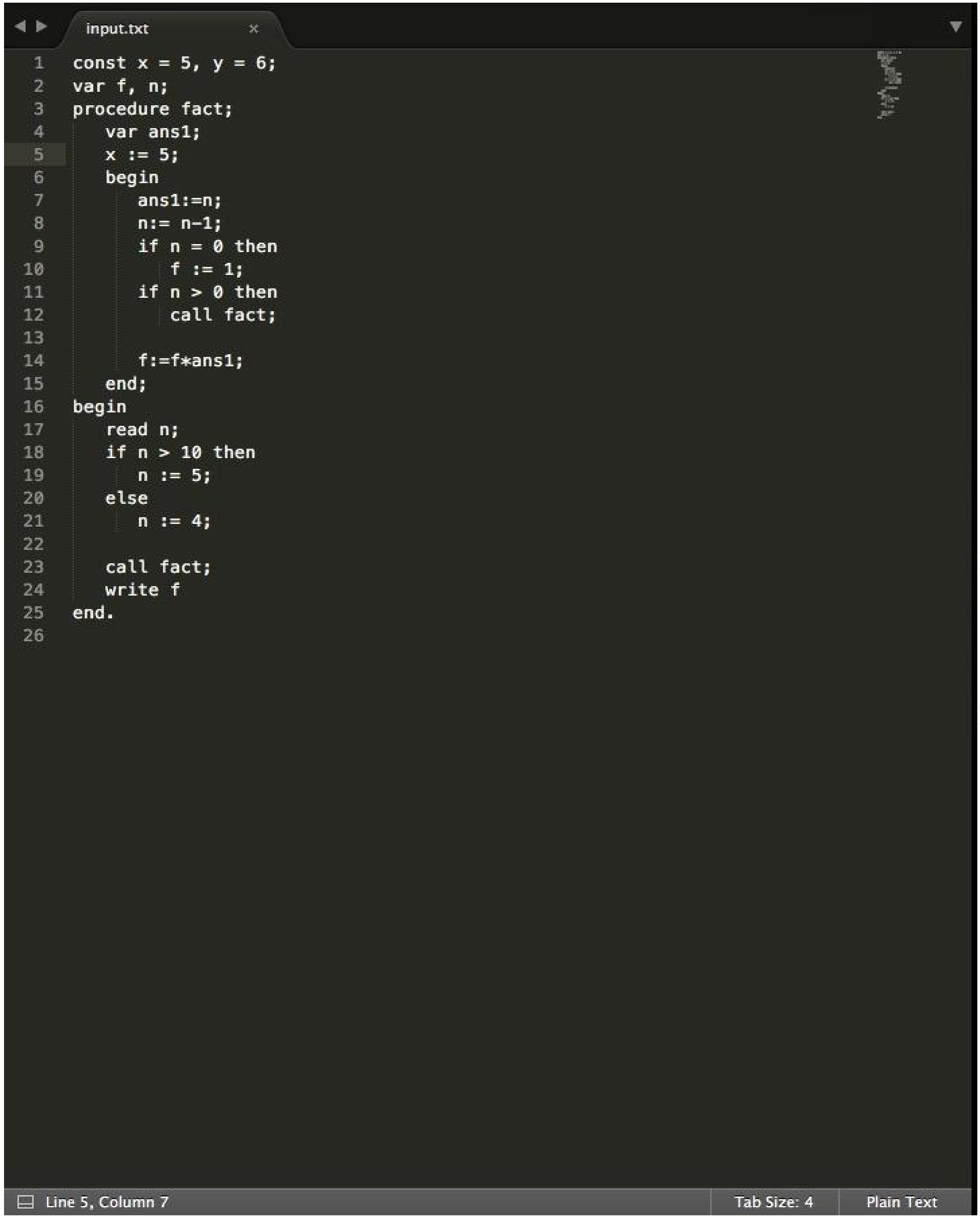


```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: f  2  0
symbol: n  2  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
29 2 f 17 2 n 18 28 2 x 9 3 5 17 2 y 9 3 6 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 2
1 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

varsym identsym f commasym identsym n semicolonsym constsym identsym x eqsym numbersym 5 commasym ide
ntsym y eqsym numbersym 6 semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n min
ussym numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbe
rsym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym id
entsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsy
m identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersy
m 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicol
onsym writesym identsym f endsym periodsym

Error 16: constants must be declared first.
Michaels-MacBook-Pro-2:Desktop mike_b$
```

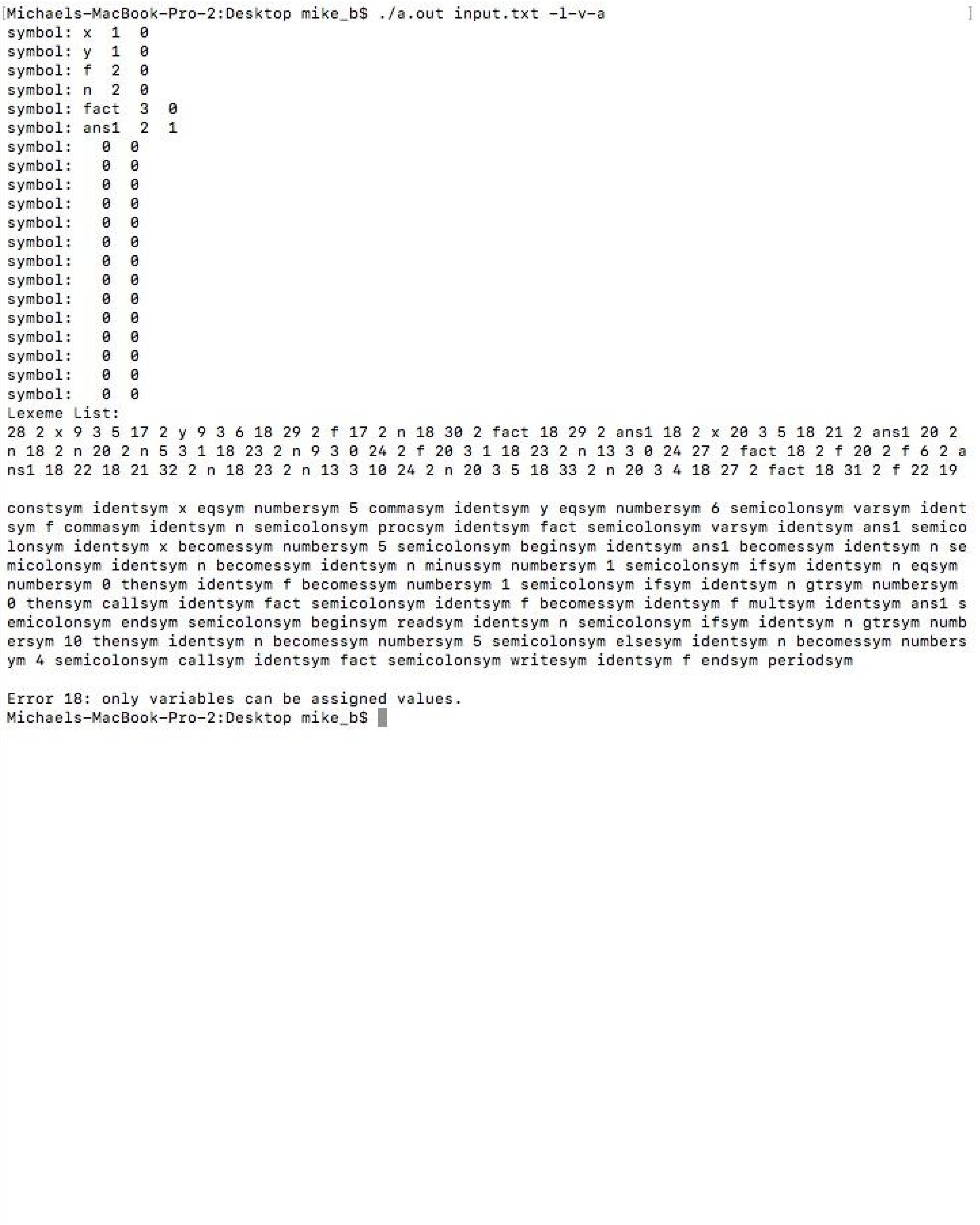


```
const x = 5, y = 6;
var f, n;
procedure fact;
    var ans1;
    begin
        ans1:=n;
        n = n-1;
        if n = 0 then
            f := 1;
        if n > 0 then
            call fact;

        f:=f*ans1;
    end;
begin
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```

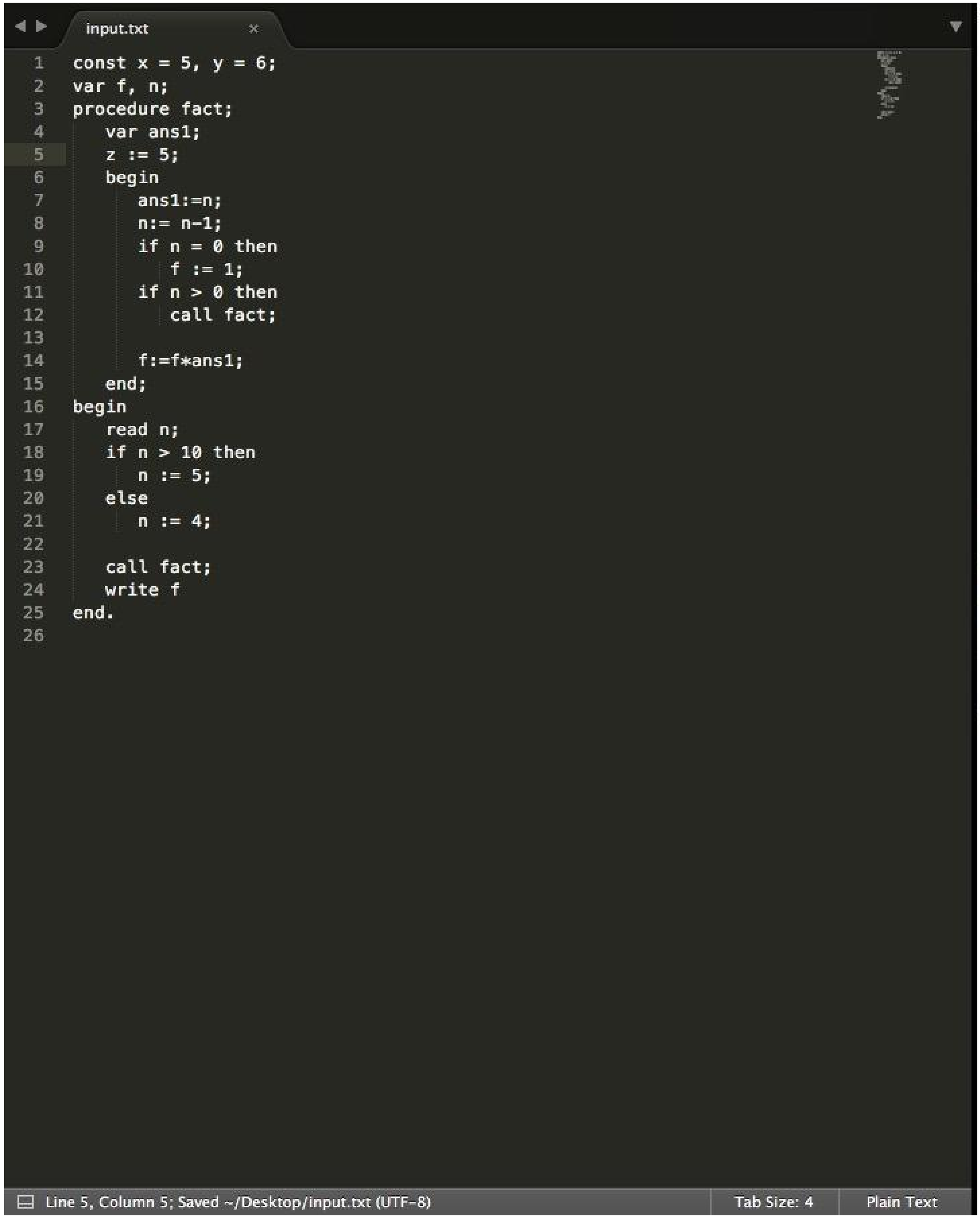
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 9 2 n
 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21
 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n eqsym identsym n minussym
 numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 thensym identsym f becomessym numbersym
1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsy
m f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym ide
ntsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 s
emicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym
 writesym identsym f endsym periodsym

Error 17: assignment symbol must follow variable.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



```
const x = 5, y = 6;
var f, n;
procedure fact;
    var ans1;
    x := 5;
    begin
        ans1:=n;
        n:= n-1;
        if n = 0 then
            f := 1;
        if n > 0 then
            call fact;

        f:=f*ans1;
    end;
begin
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```

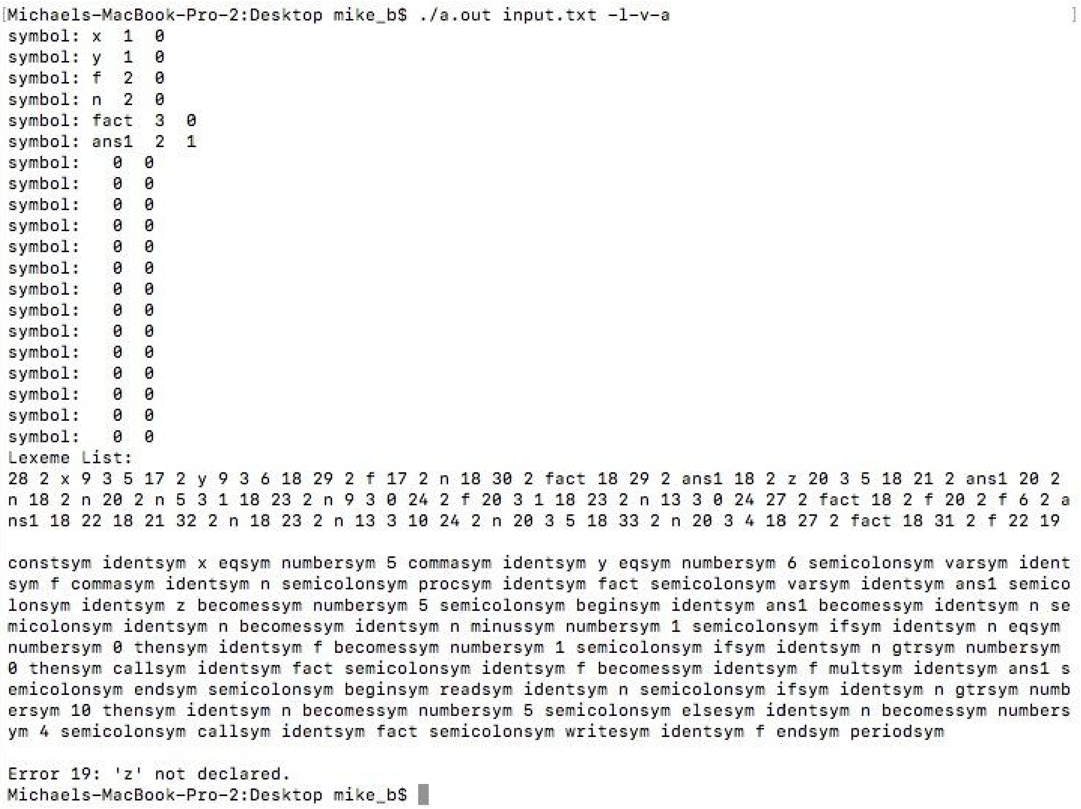


```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 2 x 20 3 5 18 21 2 ans1 20 2
n 18 2 n 20 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 a
ns1 18 22 18 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym identsym x becomessym numbersym 5 semicolonsym beginsym identsym ans1 becomessym identsym n se
micolonsym identsym n becomessym identsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym
numbersym 0 thensym identsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym
0 thensym callsym identsym fact semicolonsym identsym f becomessym identsym f multsym identsym ans1 s
emicolonsym endsym semicolonsym beginsym readsym identsym n semicolonsym ifsym identsym n gtrsym numb
ersym 10 thensym identsym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbers
ym 4 semicolonsym callsym identsym fact semicolonsym writesym identsym f endsym periodsym

Error 18: only variables can be assigned values.
Michaels-MacBook-Pro-2:Desktop mike_b$
```

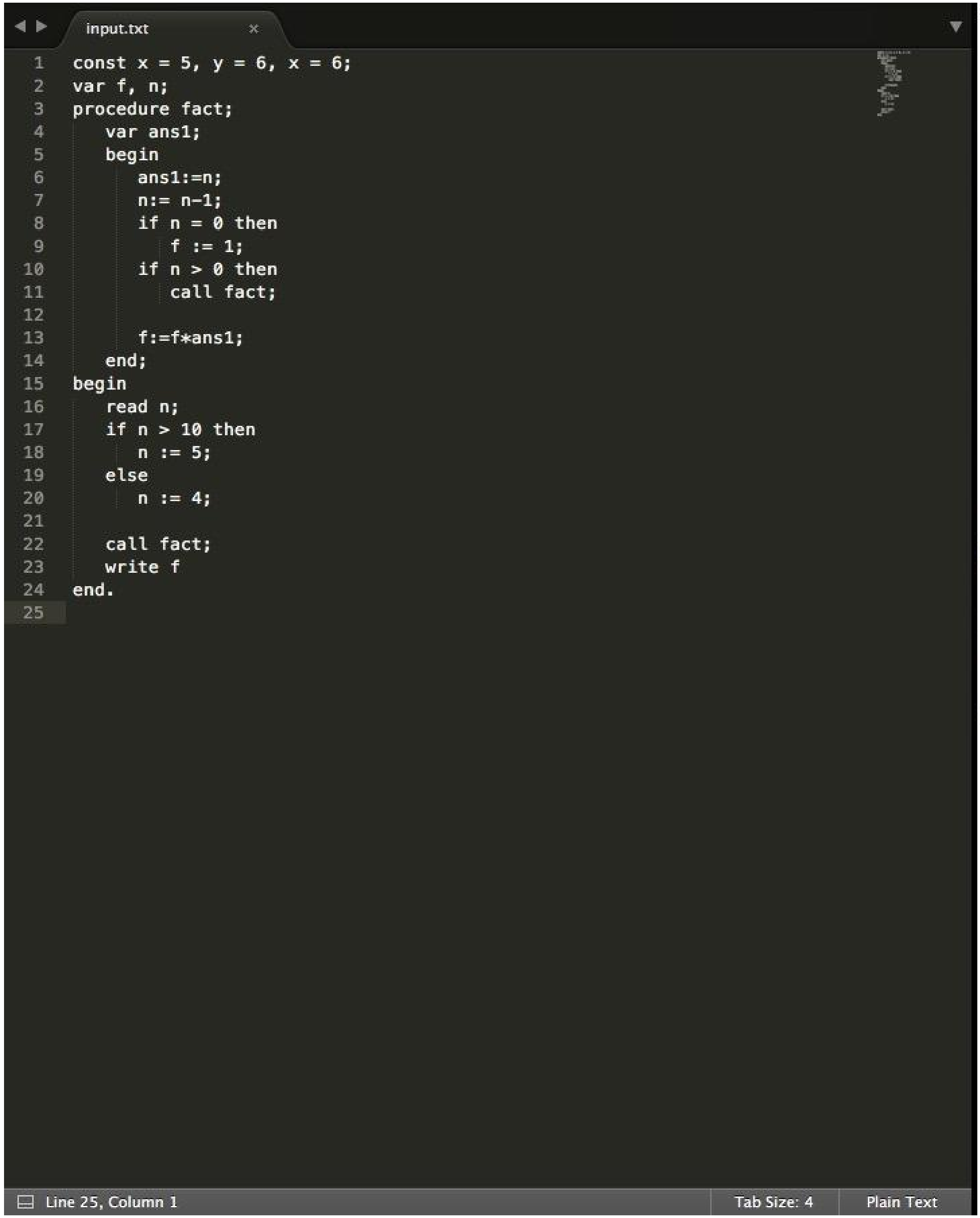


```
const x = 5, y = 6;
var f, n;
procedure fact;
   var ans1;
   z := 5;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f
end.
```

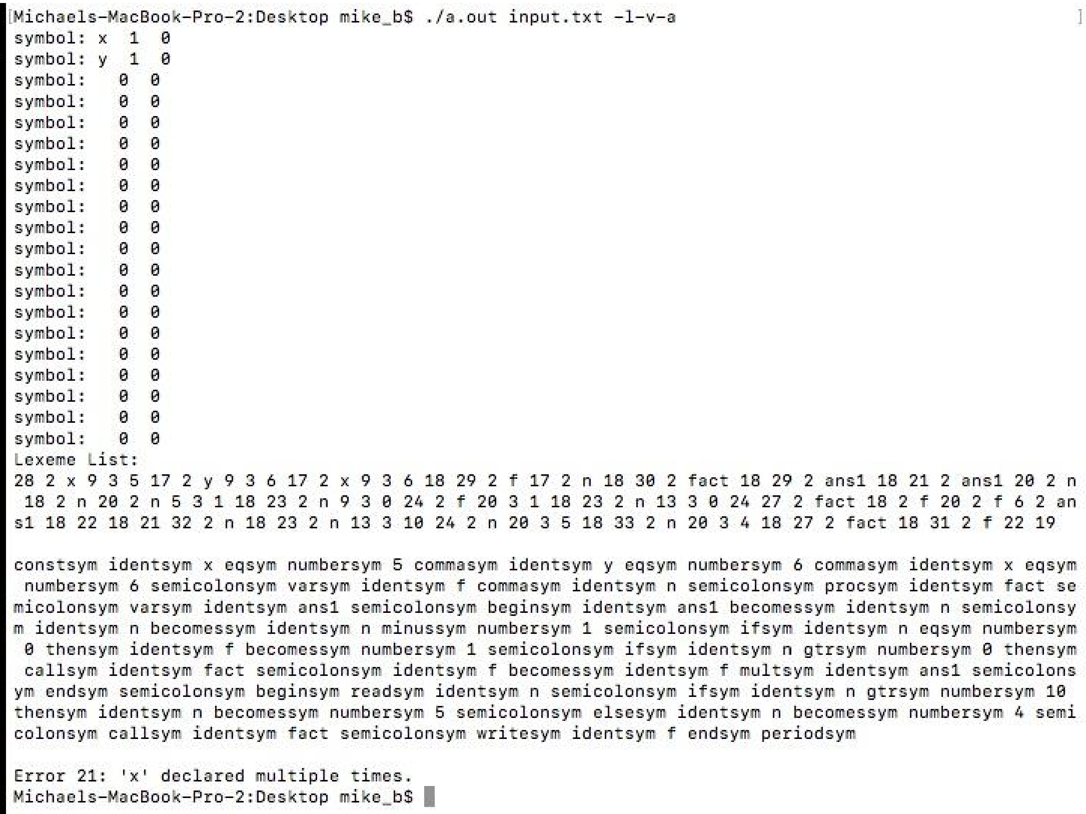
Line 5, Column 5; Saved ~/Desktop/input.txt (UTF-8)   Tab Size: 4   Plain Text



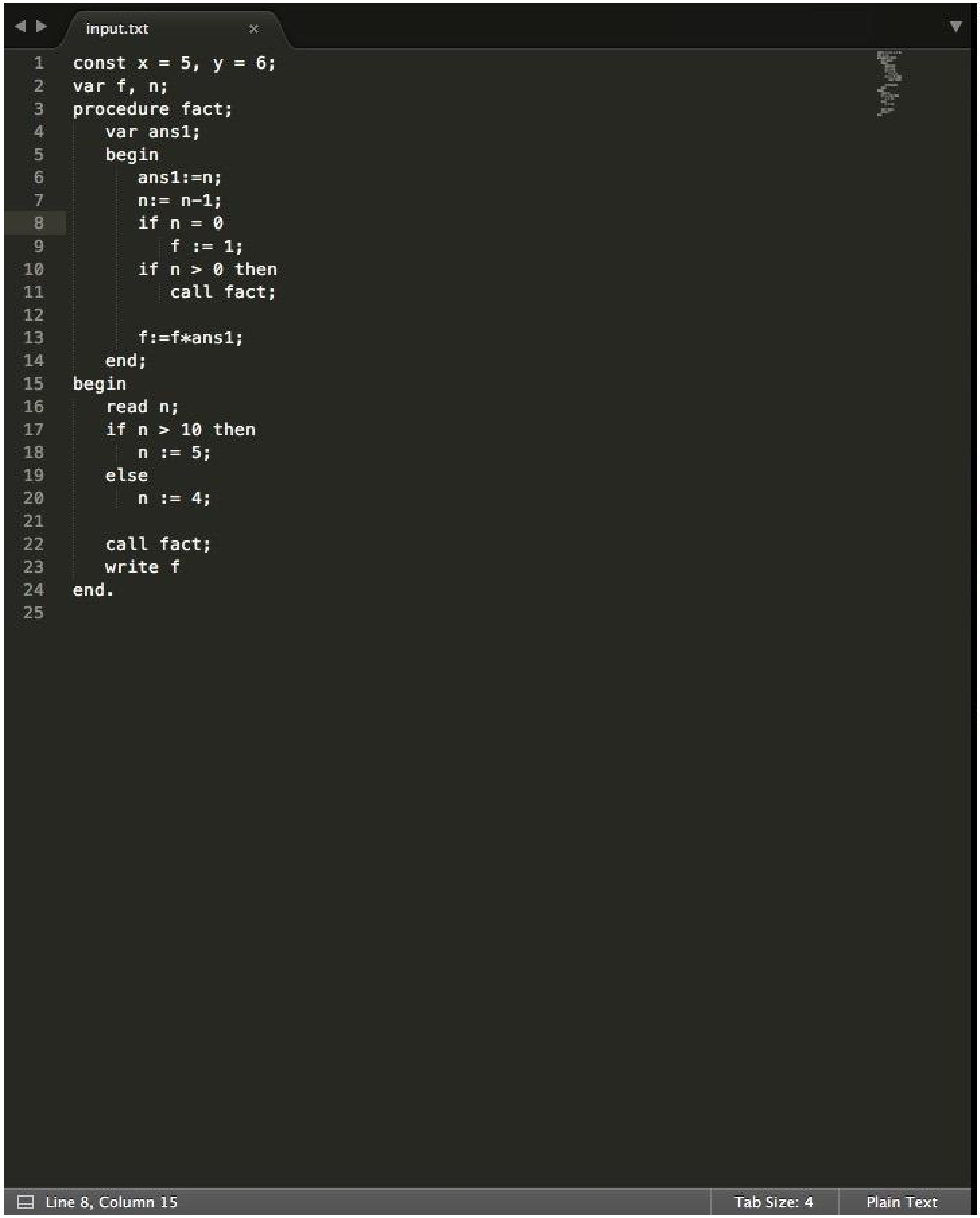
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 2 z 20 3 5 18 21 2 ans1 20 2
n 18 2 n 20 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 a
ns1 18 22 18 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym identsym z becomessym numbersym 5 semicolonsym beginsym identsym ans1 becomessym identsym n se
micolonsym identsym n becomessym identsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym
numbersym 0 thensym identsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym
0 thensym callsym identsym fact semicolonsym identsym f becomessym identsym f multsym identsym ans1 s
emicolonsym endsym semicolonsym beginsym readsym identsym n semicolonsym ifsym identsym n gtrsym numb
ersym 10 thensym identsym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbers
ym 4 semicolonsym callsym identsym fact semicolonsym writesym identsym f endsym periodsym

Error 19: 'z' not declared.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



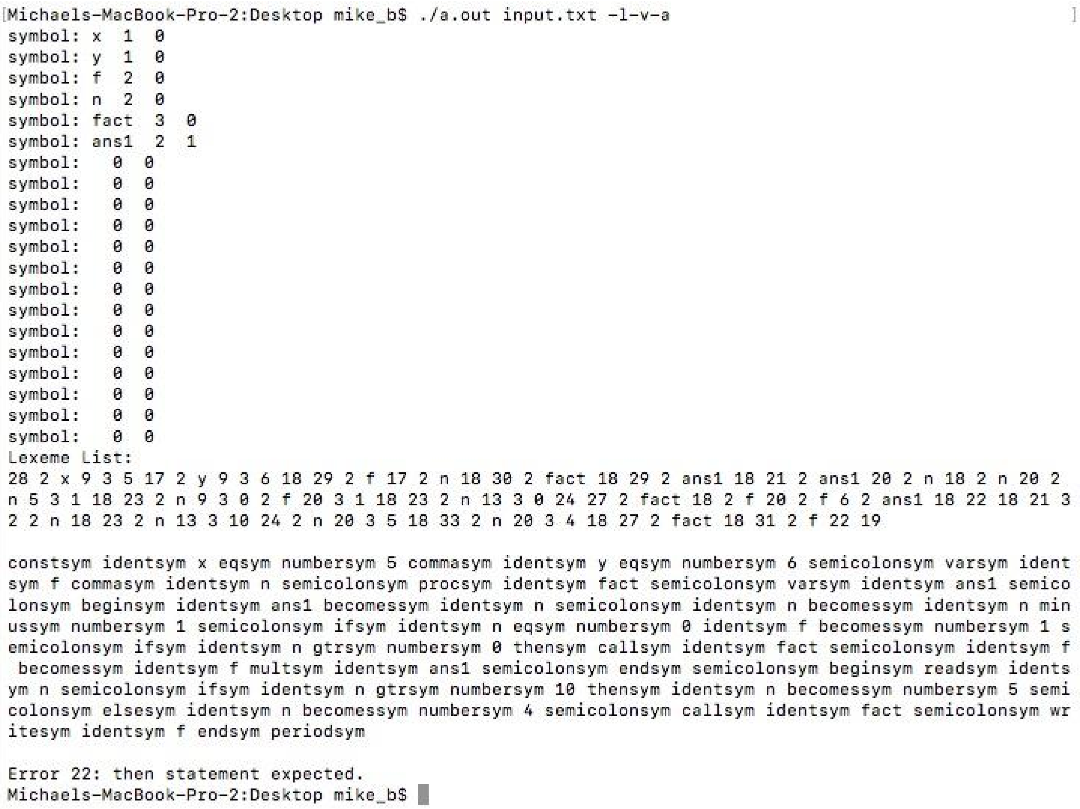
```
const x = 5, y = 6, x = 6;
var f, n;
procedure fact;
    var ans1;
    begin
        ans1:=n;
        n:= n-1;
        if n = 0 then
            f := 1;
        if n > 0 then
            call fact;

        f:=f*ans1;
    end;
begin
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```



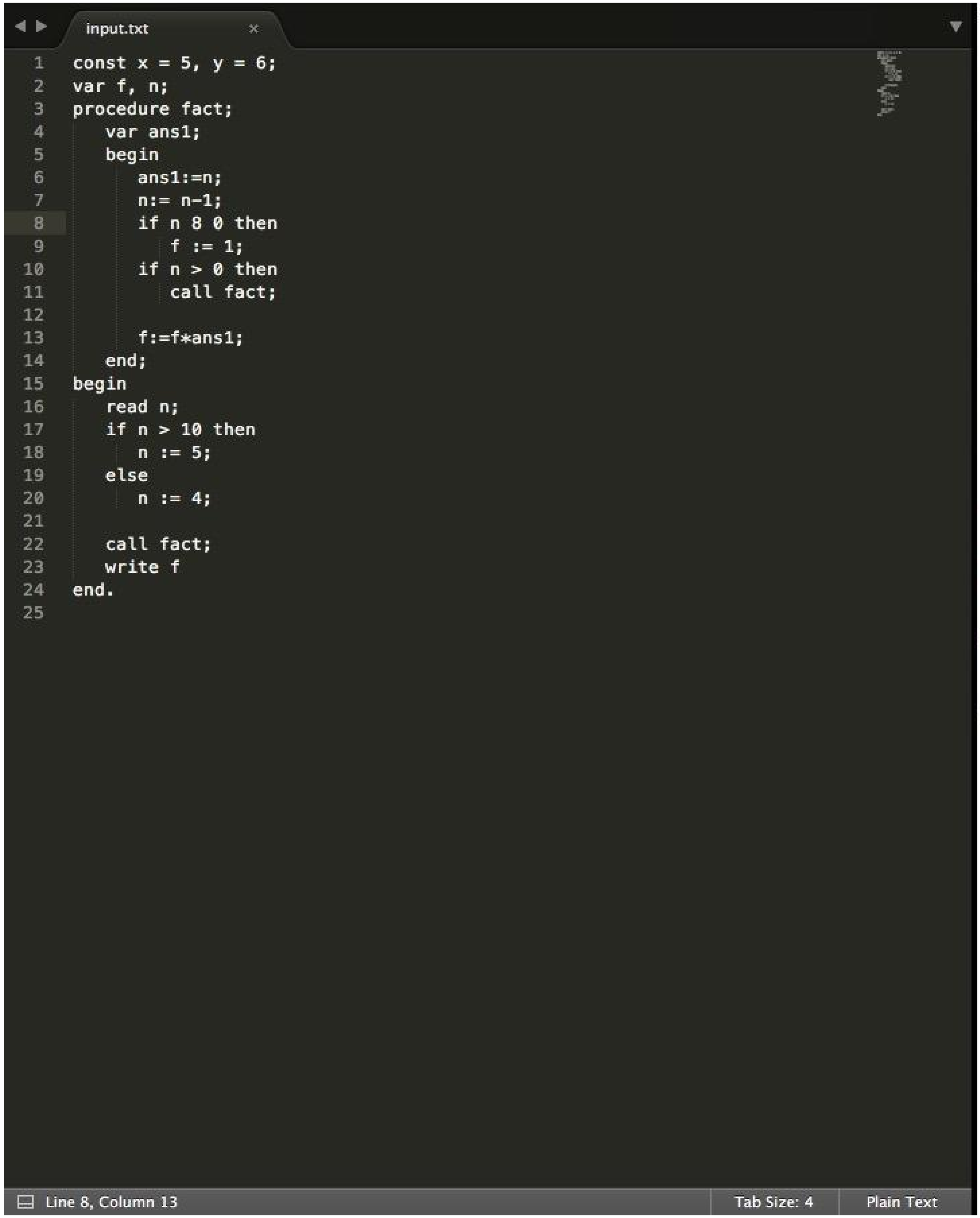
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 17 2 x 9 3 6 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n
 18 2 n 20 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 an
s1 18 22 18 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 commasym identsym x eqsym
 numbersym 6 semicolonsym varsym identsym f commasym identsym n semicolonsym procsym identsym fact se
micolonsym varsym identsym ans1 semicolonsym beginsym identsym ans1 becomessym identsym n semicolonsy
m identsym n becomessym identsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym numbersym
 0 thensym identsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym
 callsym identsym fact semicolonsym identsym f becomessym identsym f multsym identsym ans1 semicolons
ym endsym semicolonsym beginsym readsym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10
thensym identsym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semi
colonsym callsym identsym fact semicolonsym writesym identsym f endsym periodsym

Error 21: 'x' declared multiple times.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



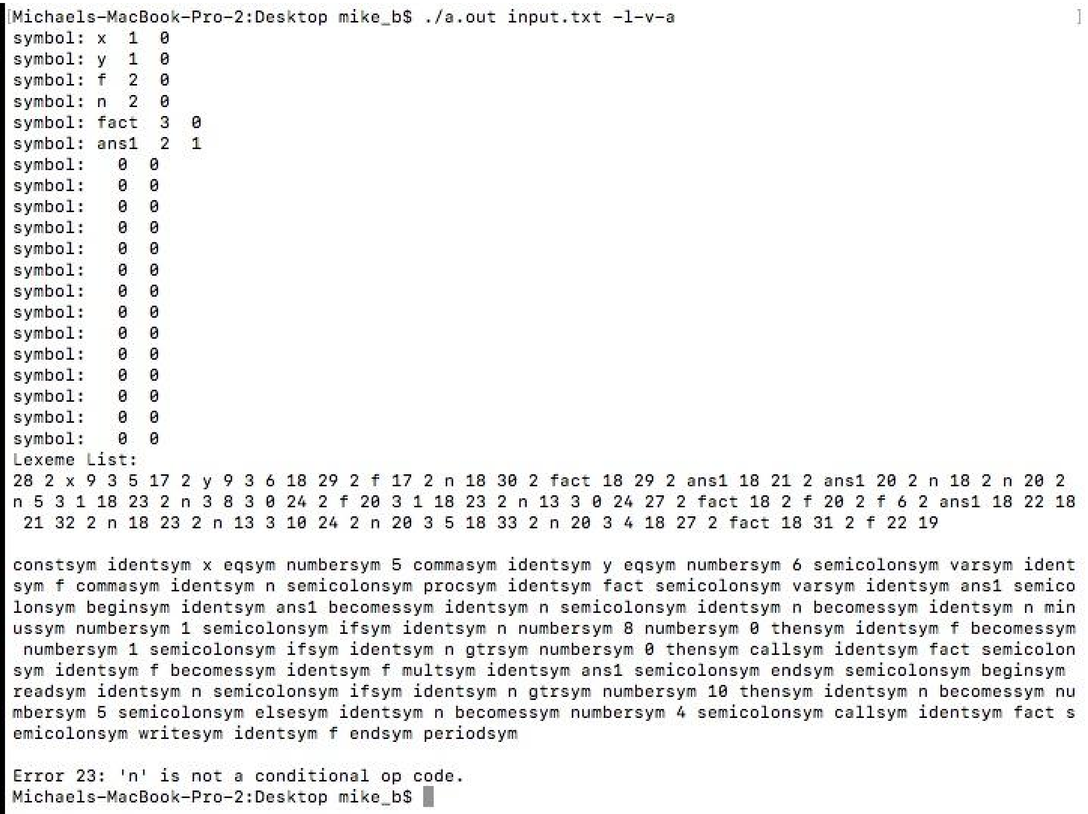
```
const x = 5, y = 6;
var f, n;
procedure fact;
    var ans1;
    begin
        ans1:=n;
        n:= n-1;
        if n = 0
            f := 1;
        if n > 0 then
            call fact;

        f:=f*ans1;
    end;
begin
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```



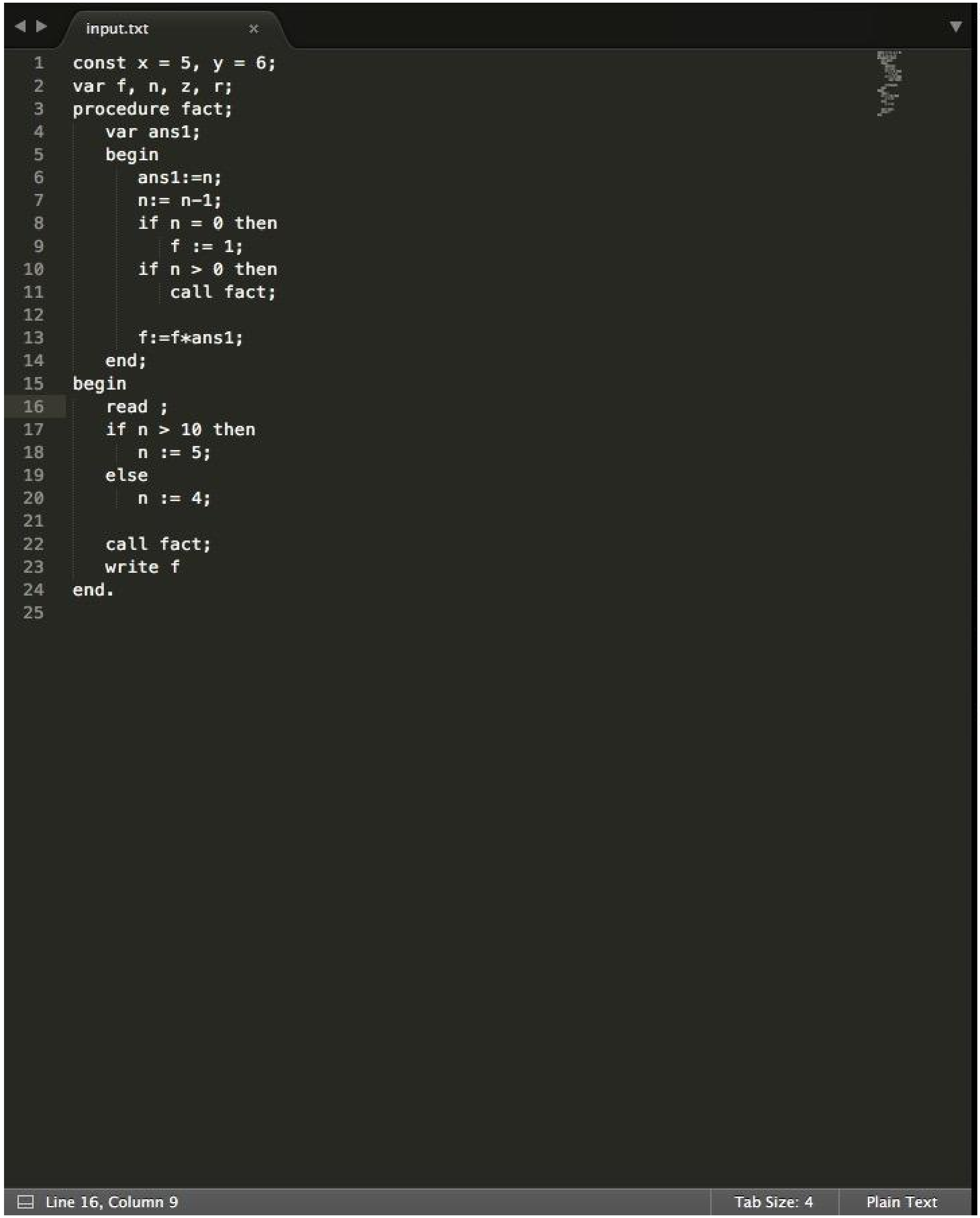
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 5 3 1 18 23 2 n 9 3 0 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18 21 3
2 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n min
ussym numbersym 1 semicolonsym ifsym identsym n eqsym numbersym 0 identsym f becomessym numbersym 1 s
emicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolonsym identsym f
 becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym readsym idents
ym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym numbersym 5 semi
colonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact semicolonsym wr
itesym identsym f endsym periodsym

Error 22: then statement expected.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



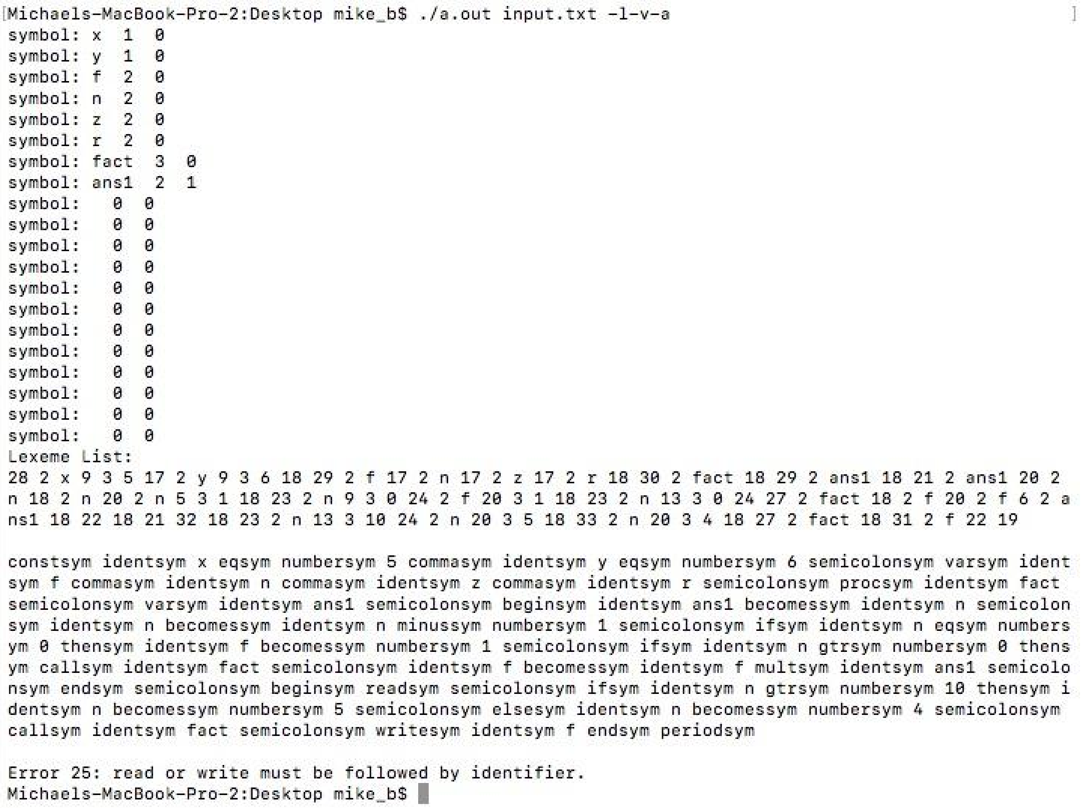
```
const x = 5, y = 6;
var f, n;
procedure fact;
    var ans1;
    begin
        ans1:=n;
        n:= n-1;
        if n 8 0 then
            f := 1;
        if n > 0 then
            call fact;

        f:=f*ans1;
    end;
begin
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```



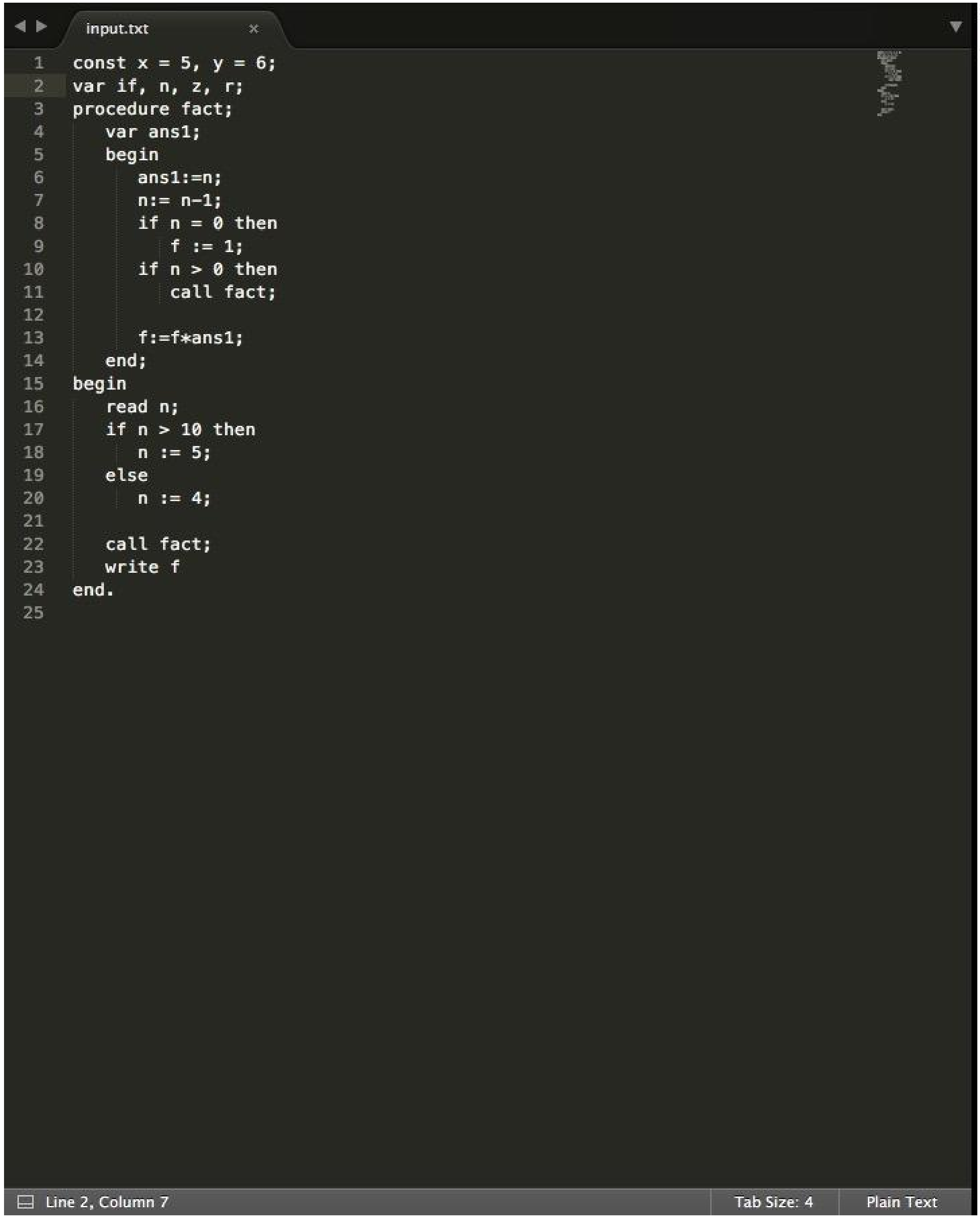
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2 n 18 2 n 20 2
n 5 3 1 18 23 2 n 3 8 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 ans1 18 22 18
 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n semicolonsym procsym identsym fact semicolonsym varsym identsym ans1 semico
lonsym beginsym identsym ans1 becomessym identsym n semicolonsym identsym n becomessym identsym n min
ussym numbersym 1 semicolonsym ifsym identsym n numbersym 8 numbersym 0 thensym identsym f becomessym
 numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thensym callsym identsym fact semicolon
sym identsym f becomessym identsym f multsym identsym ans1 semicolonsym endsym semicolonsym beginsym
readsym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thensym identsym n becomessym nu
mbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym callsym identsym fact s
emicolonsym writesym identsym f endsym periodsym

Error 23: 'n' is not a conditional op code.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



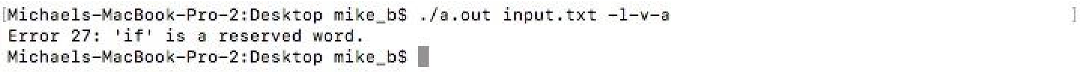
```
const x = 5, y = 6;
var f, n, z, r;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call fact;

      f:=f*ans1;
   end;
begin
   read ;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f
end.
```



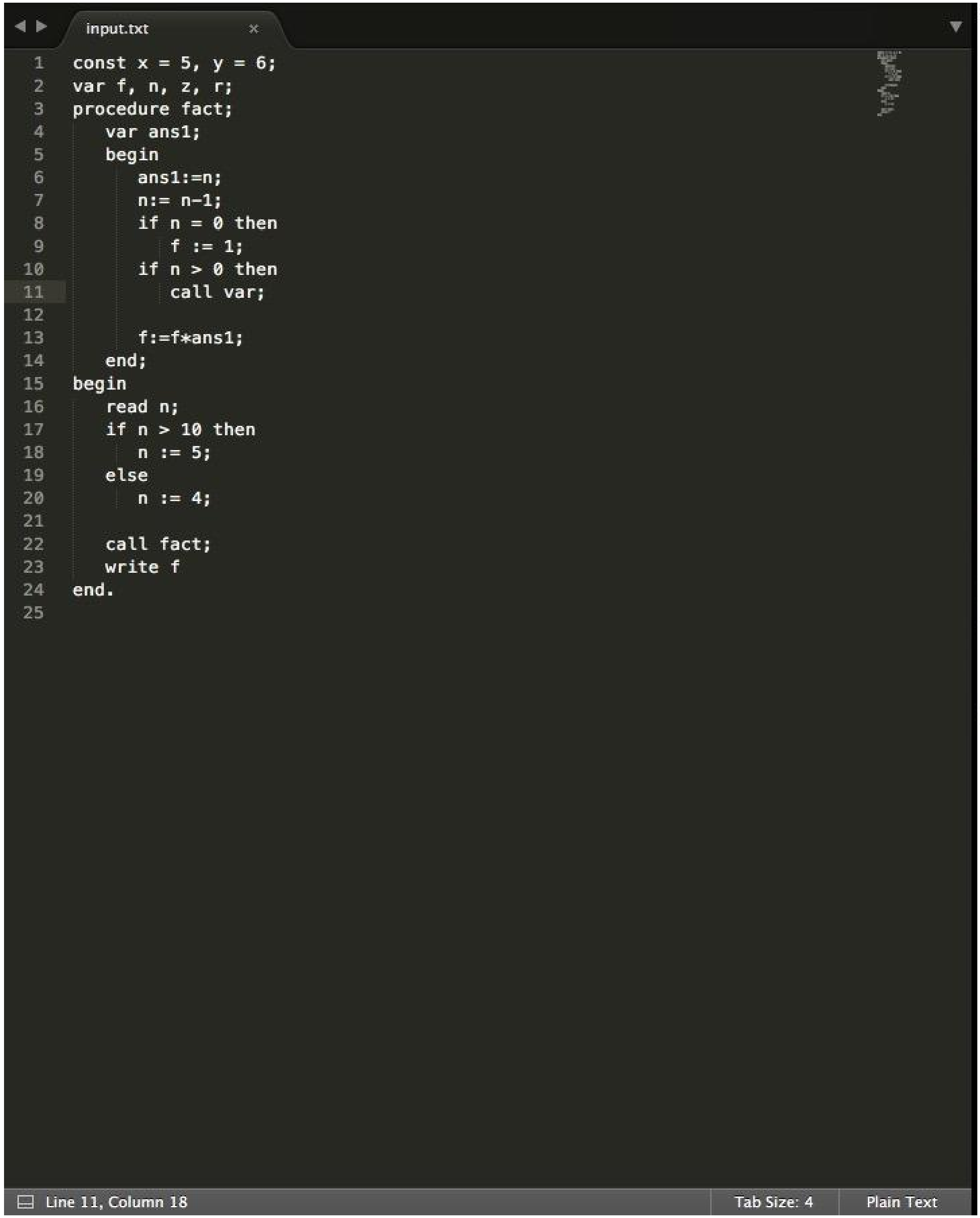
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: z  2  0
symbol: r  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 17 2 z 17 2 r 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2
n 18 2 n 20 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 fact 18 2 f 20 2 f 6 2 a
ns1 18 22 18 21 32 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n commasym identsym z commasym identsym r semicolonsym procsym identsym fact
semicolonsym varsym identsym ans1 semicolonsym beginsym identsym ans1 becomessym identsym n semicolon
sym identsym n becomessym identsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym numbers
ym 0 thensym identsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thens
ym callsym identsym fact semicolonsym identsym f becomessym identsym f multsym identsym ans1 semicolo
nsym endsym semicolonsym beginsym readsym semicolonsym ifsym identsym n gtrsym numbersym 10 thensym i
dentsym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolonsym
callsym identsym fact semicolonsym writesym identsym f endsym periodsym

Error 25: read or write must be followed by identifier.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



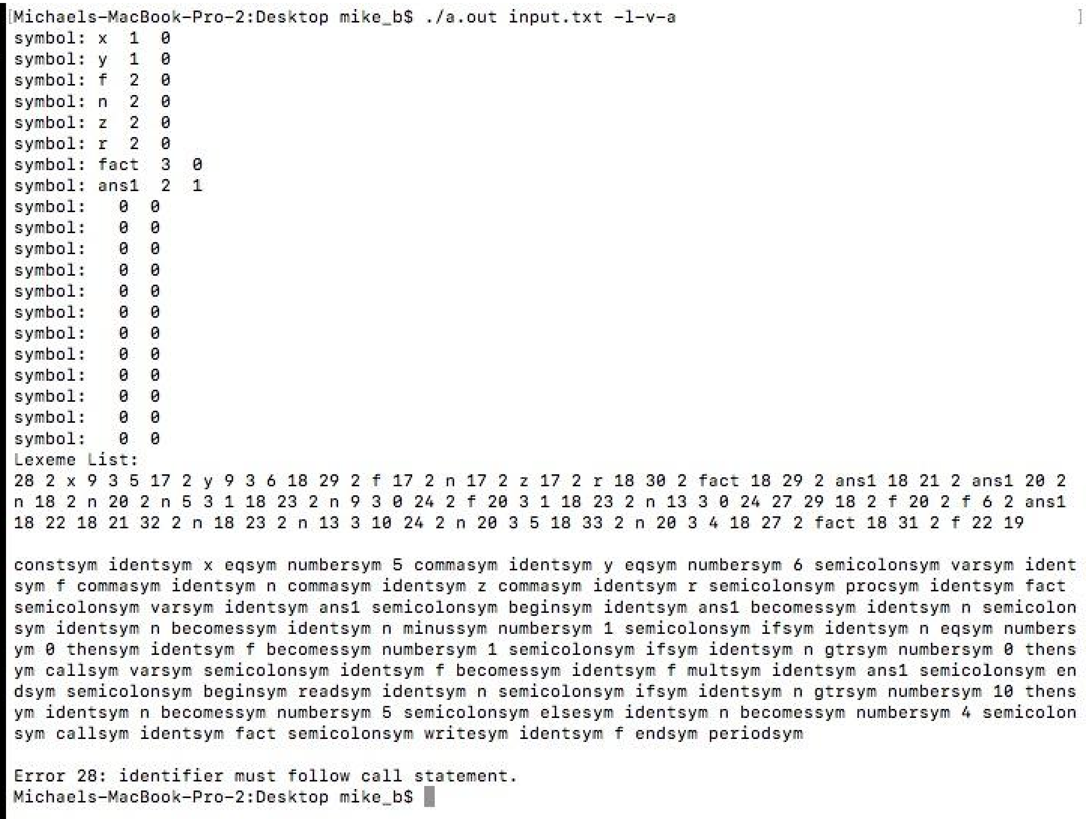
```
const x = 5, y = 6;
var if, n, z, r;
procedure fact;
    var ans1;
    begin
        ans1:=n;
        n:= n-1;
        if n = 0 then
            f := 1;
        if n > 0 then
            call fact;

        f:=f*ans1;
    end;
begin
    read n;
    if n > 10 then
        n := 5;
    else
        n := 4;

    call fact;
    write f
end.
```



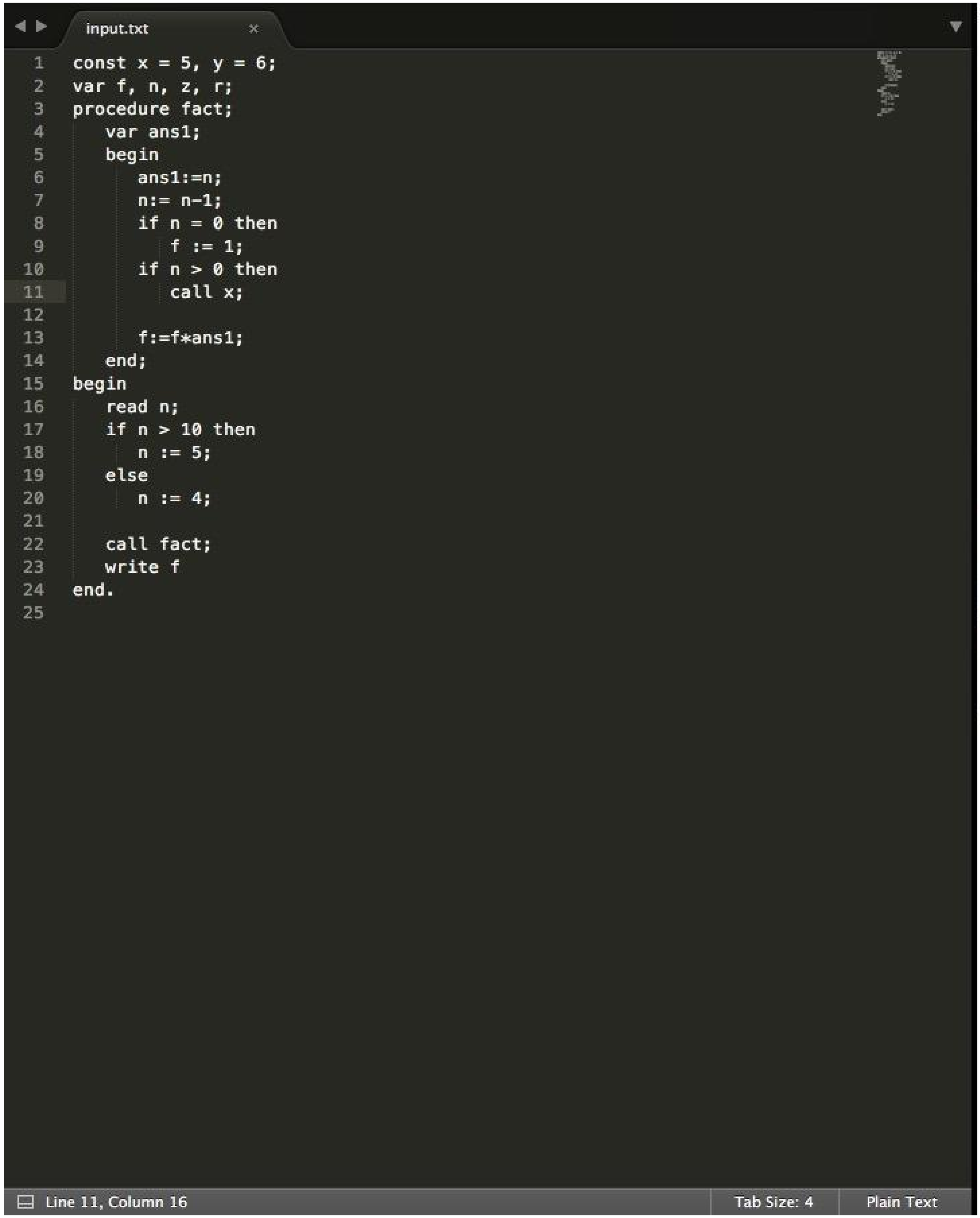
```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
Error 27: 'if' is a reserved word.
Michaels-MacBook-Pro-2:Desktop mike_b$ 
```



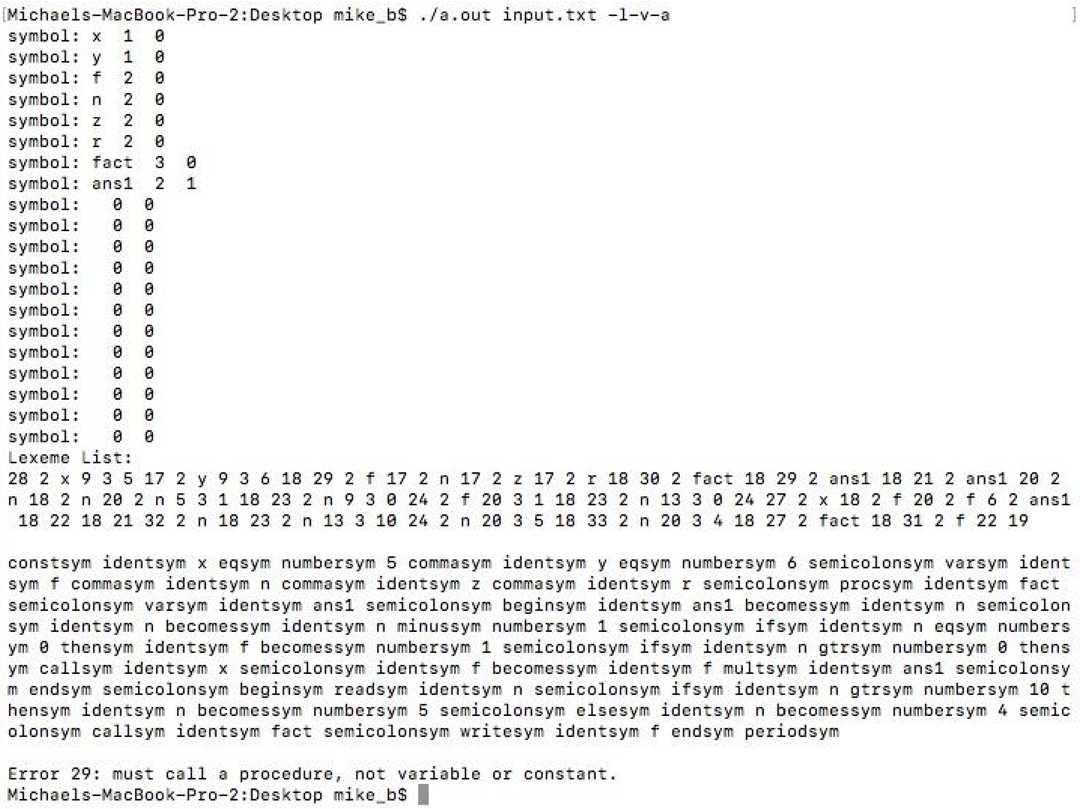
```
const x = 5, y = 6;
var f, n, z, r;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call var;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f
end.
```



```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: z  2  0
symbol: r  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 17 2 z 17 2 r 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2
n 18 2 n 20 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 29 18 2 f 20 2 f 6 2 ans1
18 22 18 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n commasym identsym z commasym identsym r semicolonsym procsym identsym fact
semicolonsym varsym identsym ans1 semicolonsym beginsym identsym ans1 becomessym identsym n semicolon
sym identsym n becomessym identsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym numbers
ym 0 thensym identsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thens
ym callsym varsym semicolonsym identsym f becomessym identsym f multsym identsym ans1 semicolonsym en
dsym semicolonsym beginsym readsym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 thens
ym identsym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semicolon
sym callsym identsym fact semicolonsym writesym identsym f endsym periodsym

Error 28: identifier must follow call statement.
Michaels-MacBook-Pro-2:Desktop mike_b$
```



```
const x = 5, y = 6;
var f, n, z, r;
procedure fact;
   var ans1;
   begin
      ans1:=n;
      n:= n-1;
      if n = 0 then
         f := 1;
      if n > 0 then
         call x;

      f:=f*ans1;
   end;
begin
   read n;
   if n > 10 then
      n := 5;
   else
      n := 4;

   call fact;
   write f
end.
```



```
Michaels-MacBook-Pro-2:Desktop mike_b$ ./a.out input.txt -l-v-a
symbol: x  1  0
symbol: y  1  0
symbol: f  2  0
symbol: n  2  0
symbol: z  2  0
symbol: r  2  0
symbol: fact  3  0
symbol: ans1  2  1
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
symbol:   0  0
Lexeme List:
28 2 x 9 3 5 17 2 y 9 3 6 18 29 2 f 17 2 n 17 2 z 17 2 r 18 30 2 fact 18 29 2 ans1 18 21 2 ans1 20 2
n 18 2 n 20 2 n 5 3 1 18 23 2 n 9 3 0 24 2 f 20 3 1 18 23 2 n 13 3 0 24 27 2 x 18 2 f 20 2 f 6 2 ans1
 18 22 18 21 32 2 n 18 23 2 n 13 3 10 24 2 n 20 3 5 18 33 2 n 20 3 4 18 27 2 fact 18 31 2 f 22 19

constsym identsym x eqsym numbersym 5 commasym identsym y eqsym numbersym 6 semicolonsym varsym ident
sym f commasym identsym n commasym identsym z commasym identsym r semicolonsym procsym identsym fact
semicolonsym varsym identsym ans1 semicolonsym beginsym identsym ans1 becomessym identsym n semicolon
sym identsym n becomessym identsym n minussym numbersym 1 semicolonsym ifsym identsym n eqsym numbers
ym 0 thensym identsym f becomessym numbersym 1 semicolonsym ifsym identsym n gtrsym numbersym 0 thens
ym callsym identsym x semicolonsym identsym f becomessym identsym f multsym identsym ans1 semicolonsy
m endsym semicolonsym beginsym readsym identsym n semicolonsym ifsym identsym n gtrsym numbersym 10 t
hensym identsym n becomessym numbersym 5 semicolonsym elsesym identsym n becomessym numbersym 4 semic
olonsym callsym identsym fact semicolonsym writesym identsym f endsym periodsym

Error 29: must call a procedure, not variable or constant.
Michaels-MacBook-Pro-2:Desktop mike_b$
```