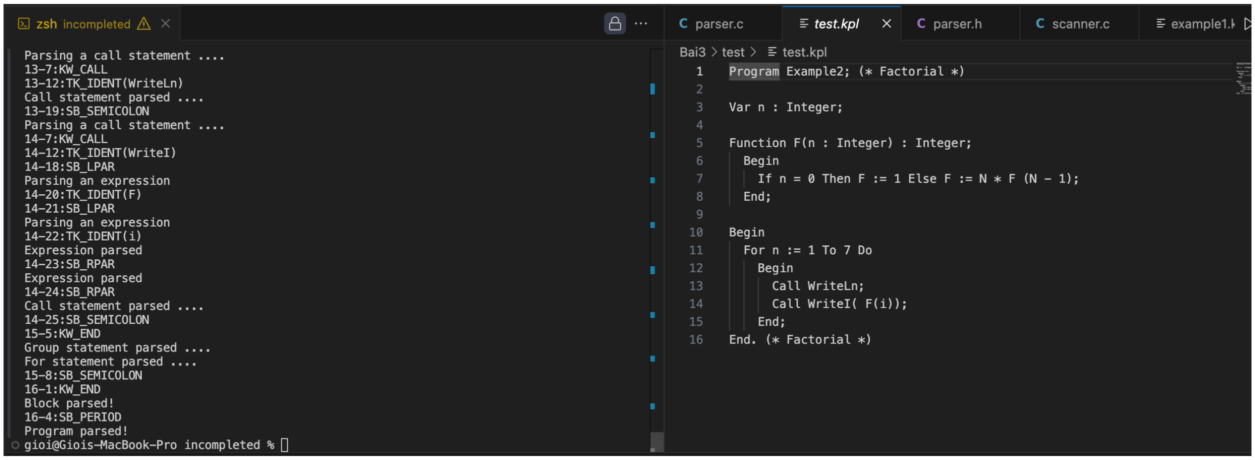
TUẦN 3: PARSER

I. Kết quả chạy

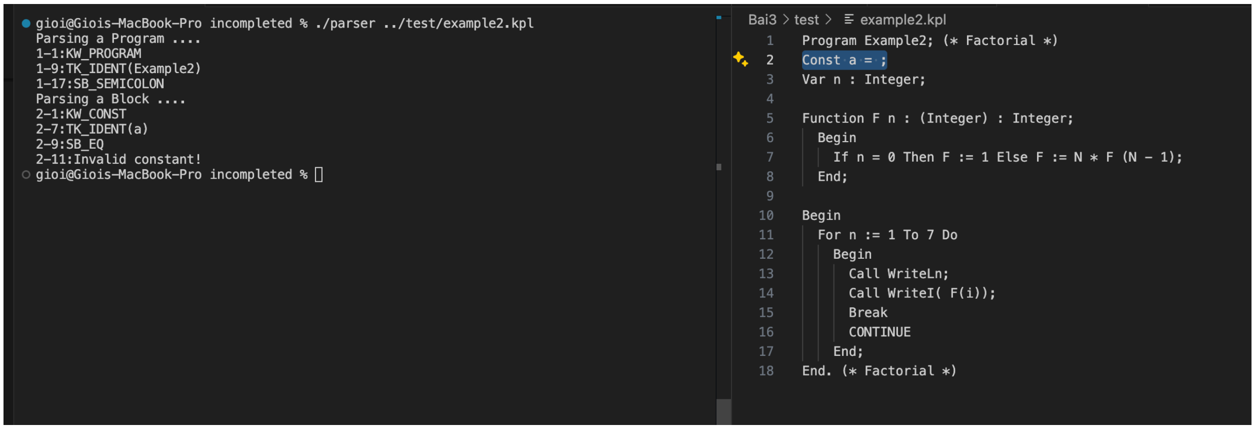


```
Parsing a call statement ....
13-7:KW_CALL
13-12:TK_IDENT(WriteLn)
Call statement parsed ....
13-19:SB_SEMICOLON
Parsing a call statement ....
14-7:KW_CALL
14-12:TK_IDENT(WriteI)
14-18:SB_LPAR
Parsing an expression
14-20:TK_IDENT(F)
14-21:SB_LPAR
Parsing an expression
14-22:TK_IDENT(i)
Expression parsed
14-23:SB_RPAR
Expression parsed
14-24:SB_RPAR
Call statement parsed ....
14-25:SB_SEMICOLON
15-5:KW_END
Group statement parsed ....
For statement parsed ....
15-8:SB_SEMICOLON
16-1:KW_END
Block parsed!
16-4:SB_PERIOD
Program parsed!
gioi@Giois-MacBook-Pro incompleted % 
```

```
Program Example2; (* Factorial *)

Var n : Integer;

Function F(n : Integer) : Integer;
  Begin
    If n = 0 Then F := 1 Else F := N * F (N - 1);
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn;
      Call WriteI( F(i));
    End;
End. (* Factorial *)
```

II. Bắt lỗi

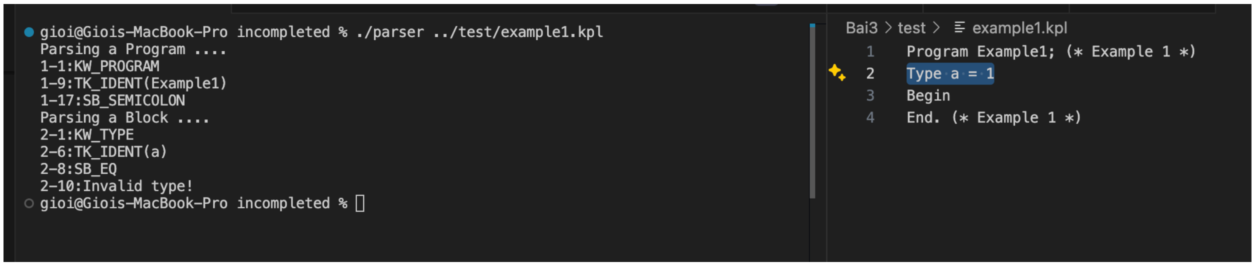
1. ERM_INVALIDCONSTANT "Invalid constant!"



```
gioi@Giois-MacBook-Pro incompleted % ./parser ../test/example2.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(Example2)
1-17:SB_SEMICOLON
Parsing a Block ....
2-1:KW_CONST
2-7:TK_IDENT(a)
2-9:SB_EQ
2-11:Invalid constant!
gioi@Giois-MacBook-Pro incompleted % 
```

```
Program Example2; (* Factorial *)
Const a = ;
Var n : Integer;

Function F n : (Integer) : Integer;
  Begin
    If n = 0 Then F := 1 Else F := N * F (N - 1);
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
End. (* Factorial *)
```

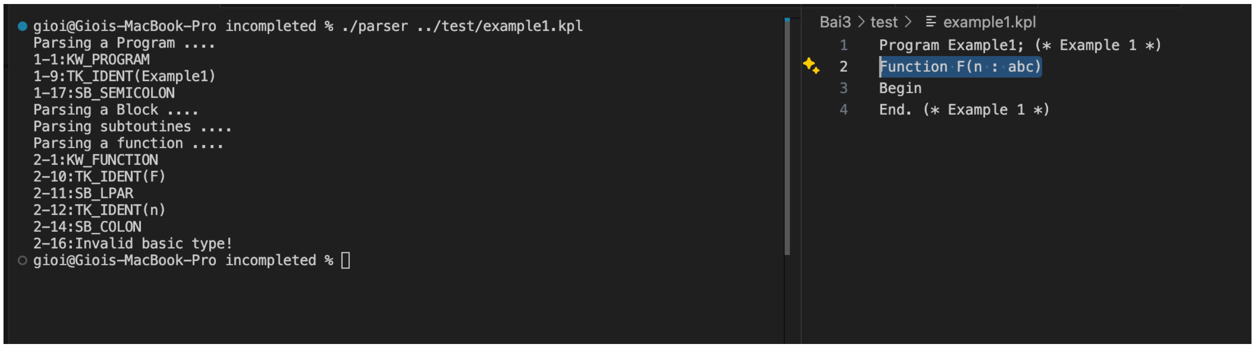
2. ERM_INVALIDTYPE "Invalid type!"



```
gioi@Giois-MacBook-Pro incompleted % ./parser ../test/example1.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(Example1)
1-17:SB_SEMICOLON
Parsing a Block ....
2-1:KW_TYPE
2-6:TK_IDENT(a)
2-8:SB_EQ
2-10:Invalid type!
gioi@Giois-MacBook-Pro incompleted % 
```

```
Program Example1; (* Example 1 *)
Type a = 1
Begin
End. (* Example 1 *)
```

3. ERM_INVALIDBASICTYPE "Invalid basic type!"

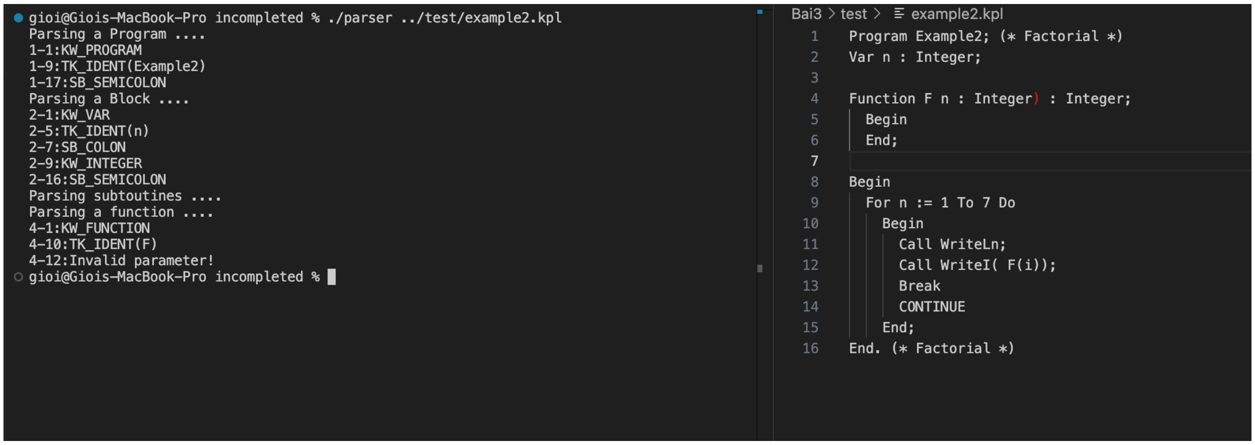


```
● gioi@Giois-MacBook-Pro incompleted % ./parser ../test/example1.kpl
  Parsing a Program ....
  1-1:KW_PROGRAM
  1-9:TK_IDENT(Example1)
  1-17:SB_SEMICOLON
  Parsing a Block ....
  Parsing subtoutines ....
  Parsing a function ....
  2-1:KW_FUNCTION
  2-10:TK_IDENT(F)
  2-11:SB_LPAR
  2-12:TK_IDENT(n)
  2-14:SB_COLON
  2-16:Invalid basic type!
○ gioi@Giois-MacBook-Pro incompleted % 
```

Bai3 > test > example1.kpl

```
Program Example1; (* Example 1 *)
Function F(n : abc)
Begin
End. (* Example 1 *)
```

4. ERM_INVALIDPARAM "Invalid parameter!"

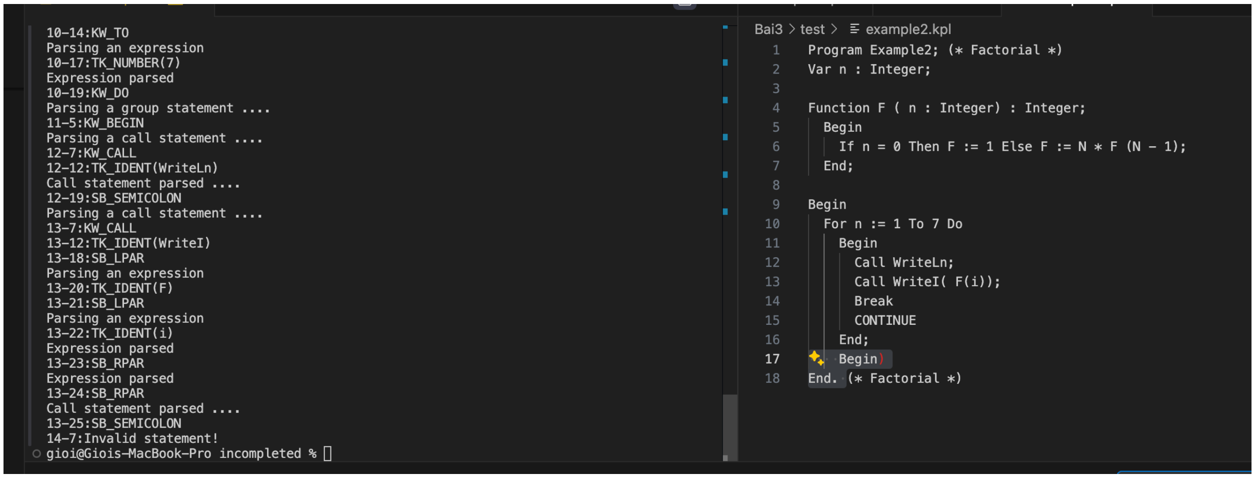


```
● gioi@Giois-MacBook-Pro incompleted % ./parser ../test/example2.kpl
  Parsing a Program ....
  1-1:KW_PROGRAM
  1-9:TK_IDENT(Example2)
  1-17:SB_SEMICOLON
  Parsing a Block ....
  2-1:KW_VAR
  2-5:TK_IDENT(n)
  2-7:SB_COLON
  2-9:KW_INTEGER
  2-16:SB_SEMICOLON
  Parsing subtoutines ....
  Parsing a function ....
  4-1:KW_FUNCTION
  4-10:TK_IDENT(F)
  4-12:Invalid parameter!
○ gioi@Giois-MacBook-Pro incompleted % 
```

Bai3 > test > example2.kpl

```
Program Example2; (* Factorial *)
Var n : Integer;

Function F n : Integer) : Integer;
  Begin
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
End. (* Factorial *)
```

5. ERM_INVALIDSTATEMENT "Invalid statement!"

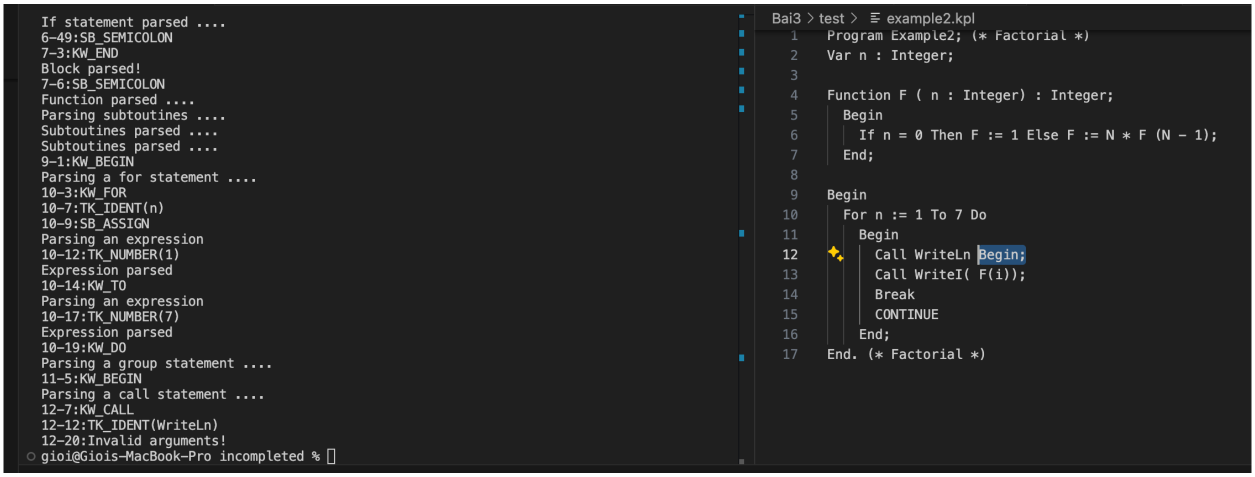


```
  10-14:KW_TO
  Parsing an expression
  10-17:TK_NUMBER(7)
  Expression parsed
  10-19:KW_DO
  Parsing a group statement ....
  11-5:KW_BEGIN
  Parsing a call statement ....
  12-7:KW_CALL
  12-12:TK_IDENT(WriteLn)
  Call statement parsed ....
  12-19:SB_SEMICOLON
  Parsing a call statement ....
  13-7:KW_CALL
  13-12:TK_IDENT(WriteI)
  13-18:SB_LPAR
  Parsing an expression
  13-20:TK_IDENT(F)
  13-21:SB_LPAR
  Parsing an expression
  13-22:TK_IDENT(i)
  Expression parsed
  13-23:SB_RPAR
  Expression parsed
  13-24:SB_RPAR
  Call statement parsed ....
  13-25:SB_SEMICOLON
  14-7:Invalid statement!
○ gioi@Giois-MacBook-Pro incompleted % 
```

Bai3 > test > example2.kpl

```
Program Example2; (* Factorial *)
Var n : Integer;

Function F ( n : Integer) : Integer;
  Begin
    If n = 0 Then F := 1 Else F := N * F (N - 1);
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
    Begin)
End. (* Factorial *)
```

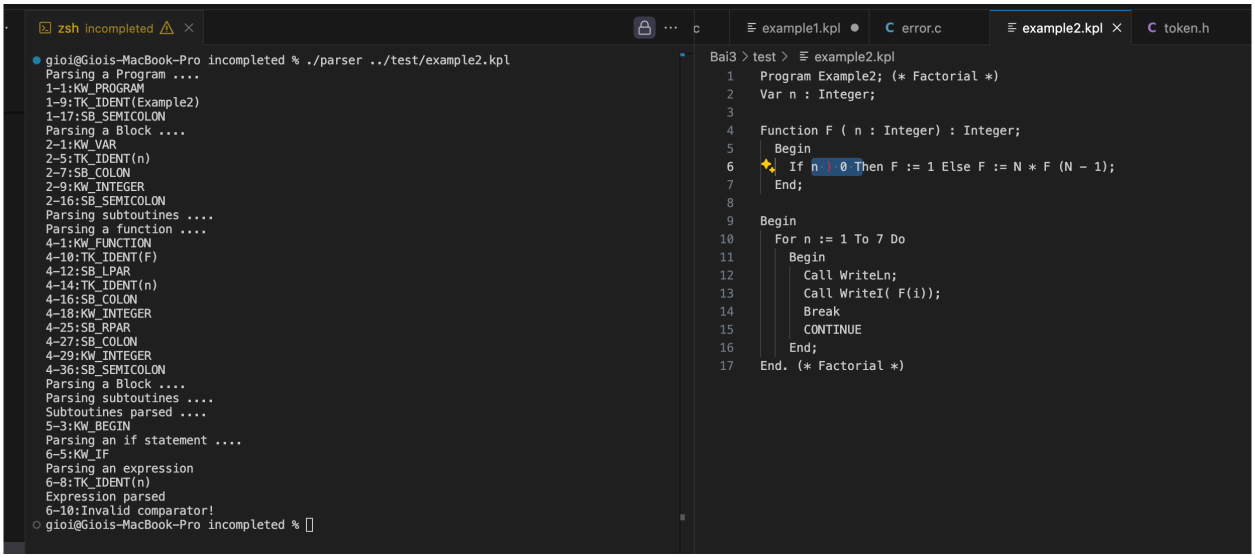
6. ERM_INVALIDARGUMENTS "Invalid arguments!"



```
If statement parsed ....
6-49:SB_SEMICOLON
7-3:KW_END
Block parsed!
7-6:SB_SEMICOLON
Function parsed ....
Parsing subtoutines ....
Subtoutines parsed ....
Subtoutines parsed ....
9-1:KW_BEGIN
Parsing a for statement ....
10-3:KW_FOR
10-7:TK_IDENT(n)
10-9:SB_ASSIGN
Parsing an expression
10-12:TK_NUMBER(1)
Expression parsed
10-14:KW_TO
Parsing an expression
10-17:TK_NUMBER(7)
Expression parsed
10-19:KW_DO
Parsing a group statement ....
11-5:KW_BEGIN
Parsing a call statement ....
12-7:KW_CALL
12-12:TK_IDENT(WriteLn)
12-20:Invalid arguments!
gioi@Giois-MacBook-Pro incompleted % 
```

```
Program Example2; (* Factorial *)
Var n : Integer;

Function F ( n : Integer) : Integer;
  Begin
    If n = 0 Then F := 1 Else F := N * F (N - 1);
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn Begin;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
End. (* Factorial *)
```

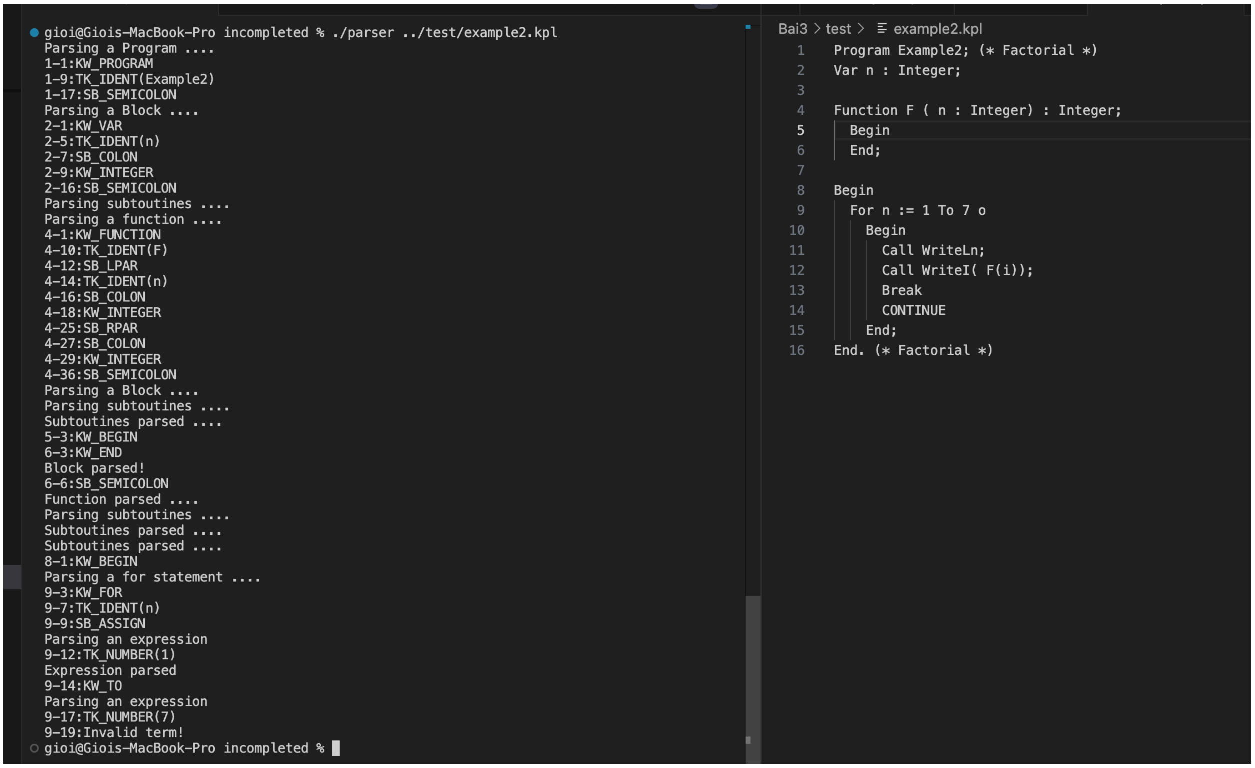
7. ERM_INVALIDCOMPARATOR "Invalid comparator!"



```
gioi@Giois-MacBook-Pro incompleted % ./parser ../test/example2.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(Example2)
1-17:SB_SEMICOLON
Parsing a Block ....
2-1:KW_VAR
2-5:TK_IDENT(n)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing subtoutines ....
Parsing a function ....
4-1:KW_FUNCTION
4-10:TK_IDENT(F)
4-12:SB_LPAR
4-14:TK_IDENT(n)
4-16:SB_COLON
4-18:KW_INTEGER
4-25:SB_RPAR
4-27:SB_COLON
4-29:KW_INTEGER
4-36:SB_SEMICOLON
Parsing a Block ....
Parsing subtoutines ....
Subtoutines parsed ....
5-3:KW_BEGIN
Parsing an if statement ....
6-5:KW_IF
Parsing an expression
6-8:TK_IDENT(n)
Expression parsed
6-10:Invalid comparator!
gioi@Giois-MacBook-Pro incompleted % 
```

```
Program Example2; (* Factorial *)
Var n : Integer;

Function F ( n : Integer) : Integer;
  Begin
    If n ) 0 Then F := 1 Else F := N * F (N - 1);
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
End. (* Factorial *)
```

8. ERM_INVALIDTERM "Invalid term!"

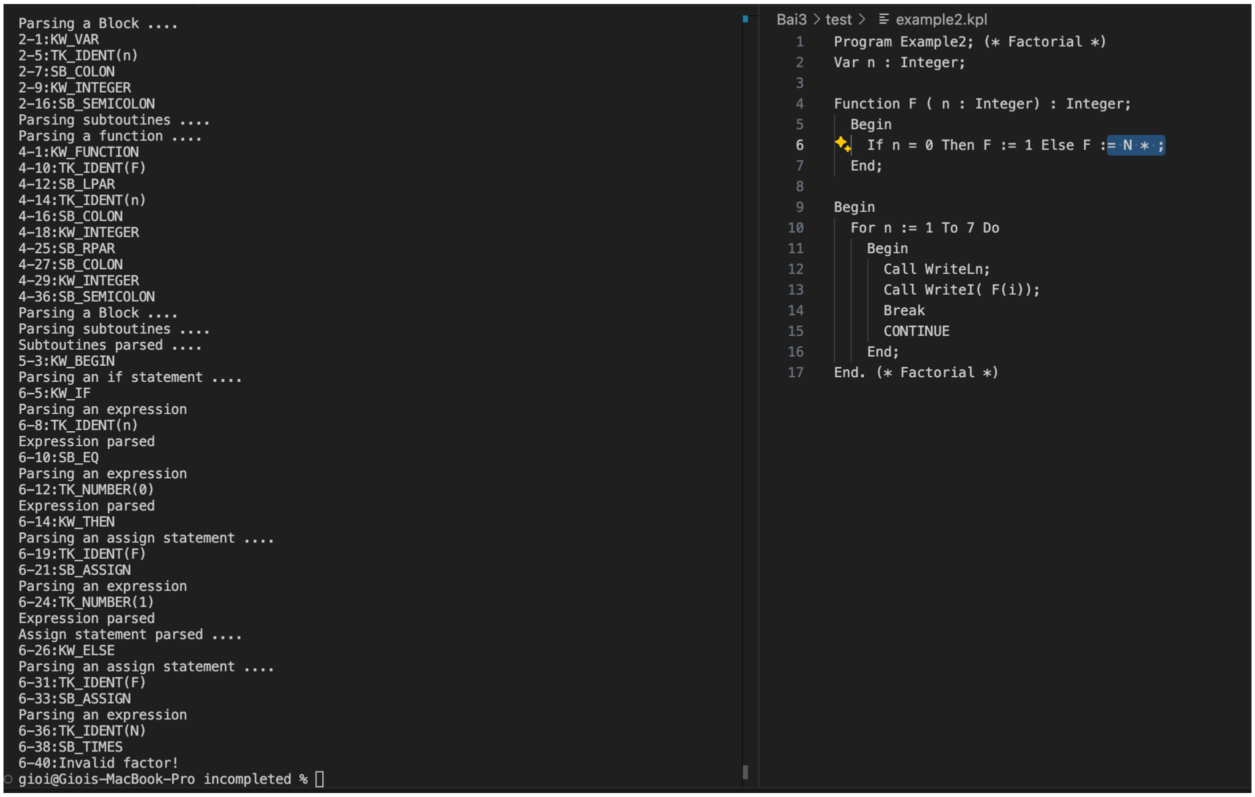


```
gioi@Giois-MacBook-Pro incompleted % ./parser ../test/example2.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(Example2)
1-17:SB_SEMICOLON
Parsing a Block ....
2-1:KW_VAR
2-5:TK_IDENT(n)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing subtoutines ....
Parsing a function ....
4-1:KW_FUNCTION
4-10:TK_IDENT(F)
4-12:SB_LPAR
4-14:TK_IDENT(n)
4-16:SB_COLON
4-18:KW_INTEGER
4-25:SB_RPAR
4-27:SB_COLON
4-29:KW_INTEGER
4-36:SB_SEMICOLON
Parsing a Block ....
Parsing subtoutines ....
Subtoutines parsed ....
5-3:KW_BEGIN
6-3:KW_END
Block parsed!
6-6:SB_SEMICOLON
Function parsed ....
Parsing subtoutines ....
Subtoutines parsed ....
Subtoutines parsed ....
8-1:KW_BEGIN
Parsing a for statement ....
9-3:KW_FOR
9-7:TK_IDENT(n)
9-9:SB_ASSIGN
Parsing an expression
9-12:TK_NUMBER(1)
Expression parsed
9-14:KW_TO
Parsing an expression
9-17:TK_NUMBER(7)
9-19:Invalid term!
gioi@Giois-MacBook-Pro incompleted %
```

Bai3 > test > example2.kpl

```
Program Example2; (* Factorial *)
Var n : Integer;

Function F ( n : Integer) : Integer;
  Begin
  End;

Begin
  For n := 1 To 7 o
    Begin
      Call WriteLn;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
End. (* Factorial *)
```

9. ERM_INVALIDFACTOR "Invalid factor!"



```
Parsing a Block ....
2-1:KW_VAR
2-5:TK_IDENT(n)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing subtoutines ....
Parsing a function ....
4-1:KW_FUNCTION
4-10:TK_IDENT(F)
4-12:SB_LPAR
4-14:TK_IDENT(n)
4-16:SB_COLON
4-18:KW_INTEGER
4-25:SB_RPAR
4-27:SB_COLON
4-29:KW_INTEGER
4-36:SB_SEMICOLON
Parsing a Block ....
Parsing subtoutines ....
Subtoutines parsed ....
5-3:KW_BEGIN
Parsing an if statement ....
6-5:KW_IF
Parsing an expression
6-8:TK_IDENT(n)
Expression parsed
6-10:SB_EQ
Parsing an expression
6-12:TK_NUMBER(0)
Expression parsed
6-14:KW_THEN
Parsing an assign statement ....
6-19:TK_IDENT(F)
6-21:SB_ASSIGN
Parsing an expression
6-24:TK_NUMBER(1)
Expression parsed
Assign statement parsed ....
6-26:KW_ELSE
Parsing an assign statement ....
6-31:TK_IDENT(F)
6-33:SB_ASSIGN
Parsing an expression
6-36:TK_IDENT(N)
6-38:SB_TIMES
6-40:Invalid factor!
gioi@Giois-MacBook-Pro incompleted %
```

Bai3 > test > example2.kpl

```
Program Example2; (* Factorial *)
Var n : Integer;

Function F ( n : Integer) : Integer;
  Begin
    If n = 0 Then F := 1 Else F := N * ;
  End;

Begin
  For n := 1 To 7 Do
    Begin
      Call WriteLn;
      Call WriteI( F(i));
      Break
      CONTINUE
    End;
End. (* Factorial *)
```