

Báo cáo LAB01

IT3323 Mã lớp 161269

Bài 1 : Phân tích từ vựng

Họ và tên: Dương Công Thuyết

Mã số sinh viên: 20225932

1. Nội dung đã sửa

2 hàm đã được modify trong code. (Token *getToken(void) và int scan(char *fileName))

2. Kết quả chạy với example1.kpl

```
duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/example1.kpl
1-1:KW_PROGRAM
1-9:TK_IDENT(Example1)
1-17:SB_SEMICOLON
2-1:KW_BEGIN
3-1:KW_END
3-4:SB_PERIOD
```

text:

1-1:KW_PROGRAM
1-9:TK_IDENT(Example1)
1-17:SB_SEMICOLON
2-1:KW_BEGIN
3-1:KW_END
3-4:SB_PERIOD

3. Kết quả chạy với example2.kpl

The screenshot shows a code editor interface with the following details:

- Left Sidebar:** Explorer, Open Editors, Scanner (containing .godo, GuiSV.Scanner, test, and several files like -sample1.kpl, -sample3.kpl, result1.txt, etc.).
- Current File:** example2.kpl
- Code Content:**

```
1 Program Example2; (* Factorial *)
2
3 Var n : Integer;
4
5 Function F(n : Integer) : Integer;
6 Begin
7     If n = 0 Then F := 1 Else F := N * F (N - 1);
8     End;
9
10 Begin
11     For n := 1 To 7 Do
12         Begin
```
- Right Panel (Analysis):** A detailed token analysis for the code. It lists tokens from 1 to 47, categorized by type:
 - 1-1: KW_PROGRAM
 - 1-2: TK_IDENT(Example2)
 - 1-3: SB_COLON
 - 3-1: KW_VAR
 - 3-2: TK_IDENT(n)
 - 3-3: SB_COLON
 - 3-4: KW_INTEGER
 - 4-1: SB_COLON
 - 5-1: KW_FUNCTION
 - 5-2: TK_IDENT(F)
 - 5-3: SB_LPAR
 - 5-4: TK_IDENT(n)
 - 5-5: SB_COLON
 - 5-6: KW_INTEGER
 - 5-7: SB_COLON
 - 5-8: KW_FUNCTION
 - 5-9: TK_IDENT(F)
 - 5-10: SB_ASSIGN
 - 5-11: TK_IDENT(n)
 - 5-12: SB_EQ
 - 5-13: KW_NUMBER(0)
 - 5-14: KW_THEN
 - 5-15: TK_IDENT(F)
 - 5-16: SB_ASSIGN
 - 5-17: TK_NUMBER(1)
 - 5-18: KW_ELSE
 - 5-19: TK_IDENT(F)
 - 5-20: SB_ASSIGN
 - 5-21: TK_IDENT(N)
 - 5-22: KW_LESS
 - 5-23: KW_GREATER
 - 5-24: TK_IDENT(F)
 - 5-25: SB_LPAR
 - 5-26: TK_IDENT(N)
 - 5-27: SB_MINUS
 - 5-28: KW_NUMBER(1)
- Bottom Status Bar:** Shows file paths, line numbers, and other editor status.

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar:
 - OPEN EDITORS**: example2.kpl
 - SCANNER**: .godo, GuisV.Scanner
 - test**: -sample1.kpl, -sample3.kpl, -result3.txt, error1_endofcomment.kpl, error2_idenccoolong.kpl, error3_invalidcharconstant.kpl, error4_invalidsymbol.kpl, error5_numbercoolong.kpl, example1.kpl, example2.kpl (selected), example3.kpl, result1.txt, result2.txt, result3.txt, charcode.c, charcode.h, error.c, error.h, reader.c, reader.h, scanner, scanner.c, token.c, token.h
- example2.kpl** file content:

```
1 Program Example2; (* Factorial *)
2
3 Var n : Integer;
4
5 Function F(n : Integer) : Integer;
6 Begin
7   If n = 0 Then F := 1 Else F := N * F (N - 1);
8 End;
9
10 Begin
11   For n := 1 To 7 Do
12     Begin
```
- TERMINAL** pane:

```
duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/example2.kpl
7-36-TK_IDENT(N)
7-36-SB_TIMES
7-37-TK_ASSIGN(F)
7-42-SB_LPAR
7-43-TK_IDENT(N)
7-45-SB_MINUS
7-47-TK_NUMBER(1)
7-49-SB_RPAR
8-1-SB_SEMICOLON
8-3-KW_END
9-1-SB_SEMICOLON
10-1-KW_BEGIN
11-3-KW_IF
11-7-TK_IDENT(n)
11-10-SB_ASSIGN
11-12-TK_NUMBER(1)
11-14-KW_TO
11-17-TK_NUMBER(7)
11-19-KW_DOWHILE
12-1-KW_BEGIN
13-7-KW_CALL
13-12-TK_IDENT(WriteLn)
14-1-SB_SEMICOLON
14-7-KW_CAL
14-21-TK_IDENT(WriteI)
14-3-SB_LPAR
14-20-TK_IDENT(F)
14-21-SB_LPAR
14-22-TK_IDENT(i)
14-24-SB_RPAR
14-25-SB_RPAR
15-1-SB_SEMICOLON
15-5-KW_END
16-1-SB_SEMICOLON
16-1-KW_END
16-4-SB_PERIOD
```
- STATUS BAR**: duongcongthuyet@Duongs-MacBook-Pro Scanner %

text:

1-1:KW_PROGRAM

1-9:TK_IDENT(Example2)

1-17:SB_SEMICOLON

3-1:KW_VAR

3-5:TK_IDENT(n)

3-7:SB_COLON

3-9:KW_INTEGER
4--1:SB_SEMICOLON
5-1:KW_FUNCTION
5-10:TK_IDENT(F)
5-11:SB_LPAR
5-12:TK_IDENT(n)
5-14:SB_COLON
5-16:KW_INTEGER
5-24:SB_RPAR
5-25:SB_COLON
5-27:KW_INTEGER
6-1:SB_SEMICOLON
6-3:KW_BEGIN
7-5:KW_IF
7-8:TK_IDENT(n)
7-10:SB_EQ
7-12:TK_NUMBER(0)
7-14:KW_THEN
7-19:TK_IDENT(F)
7-22:SB_ASSIGN
7-24:TK_NUMBER(1)
7-26:KW_ELSE
7-31:TK_IDENT(F)
7-34:SB_ASSIGN
7-36:TK_IDENT(N)
7-38:SB_TIMES
7-40:TK_IDENT(F)
7-42:SB_LPAR
7-43:TK_IDENT(N)
7-45:SB_MINUS
7-47:TK_NUMBER(1)
7-49:SB_RPAR
8--1:SB_SEMICOLON
8-3:KW_END
9--1:SB_SEMICOLON
10-1:KW_BEGIN
11-3:KW_FOR
11-7:TK_IDENT(n)
11-10:SB_ASSIGN
11-12:TK_NUMBER(1)
11-14:KW_TO
11-17:TK_NUMBER(7)

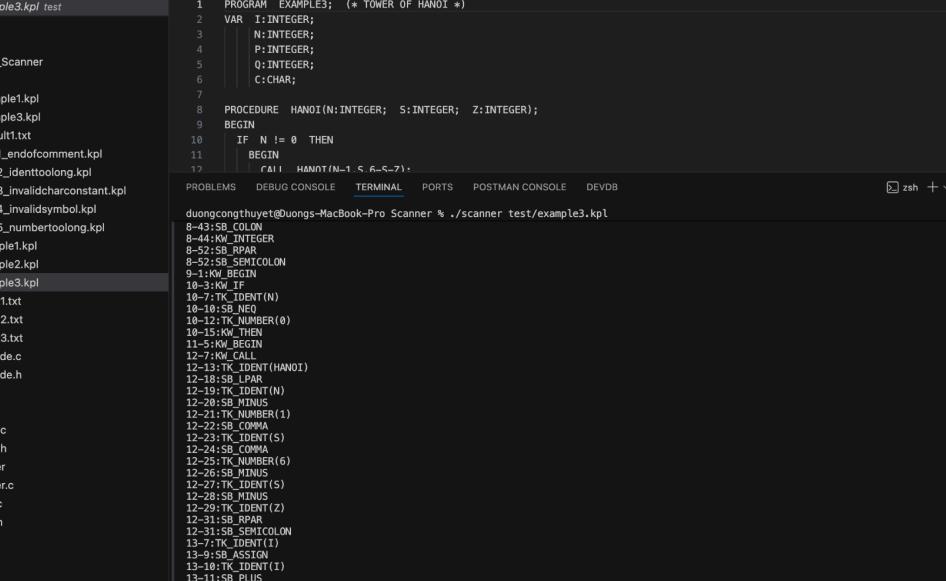
11-19:KW_DO
12-5:KW_BEGIN
13-7:KW_CALL
13-12:TK_IDENT(WriteLn)
14--1:SB_SEMICOLON
14-7:KW_CALL
14-12:TK_IDENT(WriteI)
14-18:SB_LPAR
14-20:TK_IDENT(F)
14-21:SB_LPAR
14-22:TK_IDENT(i)
14-24:SB_RPAR
14-25:SB_RPAR
15--1:SB_SEMICOLON
15-5:KW_END
16--1:SB_SEMICOLON
16-1:KW_END
16-4:SB_PERIOD

4. Kết quả chạy với example3.kpl

```
PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
VAR I:INTEGER;
N:INTEGER;
P:INTEGER;
Q:INTEGER;
C:CHAR;
BEGIN
IF N != 0 THEN
BEGIN
| CALL HANOI(N-1, S, C, T);
END;
END;
```

PROBLEMS DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE DEVDB

```
duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/example3.kpl
1-1:KW_PROGRAM
1-10:TK_IDENT(EXAMPLE3)
1-18:SB_SEMICOLON
2-1:KW_VAR
2-6:TK_IDENT(I)
2-7:SB_COLON
2-8:KW_INTEGER
2-15:SB_SEMICOLON
3-6:TK_IDENT(N)
3-7:SB_COLON
3-8:KW_INTEGER
3-15:SB_SEMICOLON
4-6:TK_IDENT(P)
4-7:SB_COLON
4-8:KW_INTEGER
4-15:SB_SEMICOLON
5-6:TK_IDENT(O)
5-7:SB_COLON
5-8:KW_INTEGER
5-15:SB_SEMICOLON
6-6:TK_IDENT(C)
6-7:SB_COLON
6-8:KW_CHAR
6-13:SB_SEMICOLON
8-14:SB_SEMICOLON
8-15:KW_PROCEDURE
8-12:TK_IDENT(HANOI)
8-17:SB_LPAR
8-18:TK_IDENT(N)
8-19:SB_COLON
8-20:KW_INTEGER
8-27:SB_SEMICOLON
8-30:TK_IDENT(S)
8-31:SB_COLON
8-32:KW_INTEGER
8-39:SB_SEMICOLON
8-42:TK_IDENT(Z)
8-43:SB_COLON
```



```
example3.kpl
test > example3.kpl
1 PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
2 VAR I:INTEGER;
3   N:INTEGER;
4   P:INTEGER;
5   Q:INTEGER;
6   C:CHAR;
7
8 PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
9 BEGIN
10   IF N != 0 THEN
11     BEGIN
12       CALL HANOT(N-1, S, Z);
13       CALL HANOT(N-1, Z, S);
14       CALL HANOT(N-1, S, Z);
15     END;
16   END;
17 END;
18
19 duongcongthuyet@Dungs-MacBook-Pro Scanner % ./scanner test/example3.kpl
8-43:SB_COLON
8-44:TK_IDENTIFIER
8-57:SB_SPACES
8-52:SB_SEMICOLON
9-1:KW_BEGIN
10-3:KW_IF
10-7:TK_IDENTIFIER(N)
10-10:SB_EQ
10-12:TK_NUMBER(0)
10-15:KW_THEN
11-5:KW_BEGIN
12-7:KW_CALL
12-11:TK_IDENTIFIER(HANOI)
13-8:SB_LPRN
12-19:TK_IDENTIFIER(N)
12-20:SB_MINUS
12-21:TK_NUMBER(1)
12-22:SB_COMMA
12-23:TK_IDENTIFIER(S)
13-8:SB_ASSIGN
12-25:TK_NUMBER(6)
12-26:SB_MINUS
12-27:TK_IDENTIFIER(S)
12-28:SB_MINUS
12-29:TK_IDENTIFIER(Z)
13-8:SB_ASSIGN
12-31:SB_SEMICOLON
13-7:TK_IDENTIFIER(I)
13-9:SB_ASSIGN
13-10:TK_IDENTIFIER(I)
13-11:SB_PLUS
13-12:TK_IDENTIFIER(1)
13-13:SB_SEMICOLON
14-7:KW_CALL
14-13:TK_IDENTIFIER(WRITELN)
14-20:SB_SEMICOLON
15-7:KW_CALL
```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar:
 - OPEN EDITORS**: example3.kpl (test)
 - SCANNER**: .godo, GuiSV_Scanner
 - test**: -sample1.kpl, -sample3.kpl, -result1.txt, error1_endofcomment.kpl, error2_identoolong.kpl, error3_invalidcharconstant.kpl, error4_invalidsymbol.kpl, error5_numbertoolong.kpl, example1.kpl, example2.kpl, example3.kpl (highlighted), result1.txt, result2.txt, result3.txt, charcode.c, error.c, error.h, reader.c, reader.h, scanner, scanner.c, token.c, token.h
- TERMINAL** tab:

```
duongcongthuyet@Oungs-MacBook-Pro Scanner % ./scanner test/example3.kpl
15-7-KW_CALL
15-13-TK_IDENT(WRITE1)
15-19-TK_LPRN
15-20-TK_IDENT(N)
16-22-SB_RPAR
16-22-SB_SEMICOLON
16-7-KW_CALL
16-13-TK_IDENT(WRITE1)
16-19-TK_LPRN
16-20-TK_IDENT(N)
16-22-SB_RPAR
16-22-SB_SEMICOLON
17-7-KW_CALL
17-13-TK_IDENT(WRITE1)
17-19-TK_LPRN
17-20-TK_IDENT(S)
17-22-SB_RPAR
17-22-SB_SEMICOLON
18-7-KW_CALL
18-13-TK_IDENT(WRITE1)
18-19-TK_LPRN
18-20-TK_IDENT(Z)
18-22-SB_RPAR
18-22-SB_SEMICOLON
19-7-KW_CALL
19-13-TK_IDENT(HANOI)
19-19-TK_LPRN
19-19-TK_IDENT(N)
19-20-SB_MINUS
19-21-TK_NUMBER(1)
19-22-SB_COMMA
19-23-TK_NUMBER(6)
19-23-SB_MINUS
19-25-TK_IDENT(S)
19-26-SB_MINUS
19-27-TK_IDENT(Z)
19-28-SB_COMMA
```
- STATUS BAR**: thuannguyen2034/web20241#17 needs reviewers, Reconnect to Discord, Ln 1, Col 1, Spaces: 2, UTF-8, CRLF, Plain Text, Go Live, Colorize: 0 variables, Colorize, Prettier

Scanning results for example3.kpl:

```
PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
VAR I:INTEGER;
N:INTEGER;
P:INTEGER;
Q:INTEGER;
C:CHAR;

PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;

HANOT(N, S, Z) = BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;

HANOT(N-1, S, C) = BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;

HANOT(N-1, C, Z) = BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;
```

Scanning results for example3.kpl (continued):

```
19-28:SB COMMA
19-29:TK IDENT(Z)
19-31:SB RPAR
20-5:KW END
21-14:KW END
21-15:SB SEMICOLON
23-1:KW BEGIN
24-3:KW FOR
24-8:TK IDENT(N)
24-11:SB ASSIGN
24-12:TK NUMBER(1)
24-16:KW TO
24-17:TK NUMBER(4)
24-20:TK NUMBER(4)
24-23:KW DO
25-5:KW BEGIN
26-7:KW FOR
26-12:TK IDENT(I)
26-13:SB ASSIGN
26-15:TK NUMBER(1)
26-18:KW TO
26-22:TK NUMBER(4)
26-25:KW DO
27-5:KW CALL
27-15:TK IDENT(WRITEC)
27-21:SB LPAR
27-23:TK CHAR(' ')
27-26:SB RPAR
27-26:SB SEMICOLON
28-7:TK IDENT(READC)
28-18:SB LPAR
28-19:TK IDENT(C)
28-21:SB RPAR
28-21:SB SEMICOLON
29-1:TK IDENT(WRITELN)
29-13:TK IDENT(WRITELN)
29-19:SB LPAR
```

Scanning results for example3.kpl:

```
PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
VAR I:INTEGER;
N:INTEGER;
P:INTEGER;
Q:INTEGER;
C:CHAR;

PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;

HANOT(N, S, Z) = BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;

HANOT(N-1, S, C) = BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;

HANOT(N-1, C, Z) = BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, C, Z);
END;
END;
```

Scanning results for example3.kpl (continued):

```
29-19:SB LPAR
29-20:TK IDENT(C)
29-22:SB RPAR
30-5:KW END
30-8:SB SEMICOLON
31-3:TK IDENT(P)
31-5:SB ASSIGN
31-6:TK NUMBER(1)
31-7:SB SEMICOLON
32-3:TK IDENT(I)
32-5:SB ASSIGN
32-6:TK NUMBER(2)
32-7:SB SEMICOLON
33-3:KW FOR
33-8:TK IDENT(N)
33-10:SB ASSIGN
33-11:TK NUMBER(2)
33-14:KW TO
33-18:TK NUMBER(4)
33-21:KW DO
34-5:KW BEGIN
35-7:TK IDENT(I)
35-9:SB ASSIGN
35-10:TK NUMBER(0)
35-11:SB SEMICOLON
36-7:KW CALL
36-13:TK IDENT(HANOI)
36-18:SB LPAR
36-19:TK IDENT(N)
36-20:SB COMMA
36-21:TK IDENT(P)
36-22:SB COMMA
36-23:TK IDENT(Q)
36-24:SB RPAR
36-25:SB SEMICOLON
37-7:KW CALL
37-13:TK IDENT(WRITELN)
```

The screenshot shows a code editor interface with the following details:

- Left Sidebar (EXPLORER):** Shows a file tree with categories like OPEN EDITORS, SCANNER, and test. The test folder contains files such as example1.kpl, example2.kpl, and example3.kpl.
- Top Bar:** Includes tabs for RECENT, RECONNECT, and DISCORD, along with a search bar labeled "Scanner".
- Central Area:** A code editor window titled "example3.kpl" containing the following KPL code:

```
PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
VAR I:INTEGER;
N:INTEGER;
P:INTEGER;
Q:INTEGER;
C:CHAR;
PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
BEGIN
IF N != 0 THEN
BEGIN
CALL HANOT(N-1, S, Z);
CALL HANOT(N-1, Z, S);
END;
END;
HANOI(3, 1, 2);
END.
```
- Bottom Status Bar:** Displays the current file path: "duongcongthuyet@Oungs-MacBook-Pro Scanner %", along with other system information like Ln 1, Col 1, Spaces: 2, UTF-8, CRLF, Plain Text, Go Live, Colorize: 0 variables, Colorize, and Prettier.

text:

1-1:KW PROGRAM

1-10:TK IDENT(EXAMPLE3)

1-18:SB SEMICOLON

2-1:KW VAR

2-6:TK IDENT(I)

2-7:SB COLON

2-8:KW INTEG

2-15:SB SEMICOL

3-6:TK IDFNT(N)

3-7-SB COLON

3-8-KW INTEG

3-15-SB SEMICON

4-6-TK IDENT(P)

4-7-SB_COLON

4-8-KW INTEGR

4-15-SB SEMICO

5-6:TK IDENT()

5-6:TR_IDENT()

5 9:KW INTECH

5-15:SB SEMICO

6-6:TK IDENT(C)

6-7:SB_COLON

6 ,ISBN_C005N

6-12:SB_SEMICOLON
8-1:KW PROCEDURE
8-12:TK_IDENT(HANOI)
8-17:SB_LPAR
8-18:TK_IDENT(N)
8-19:SB_COLON
8-20:KW_INTEGER
8-27:SB_SEMICOLON
8-30:TK_IDENT(S)
8-31:SB_COLON
8-32:KW_INTEGER
8-39:SB_SEMICOLON
8-42:TK_IDENT(Z)
8-43:SB_COLON
8-44:KW_INTEGER
8-52:SB_RPAR
8-52:SB_SEMICOLON
9-1:KW_BEGIN
10-3:KW_IF
10-7:TK_IDENT(N)
10-10:SB_NEQ
10-12:TK_NUMBER(0)
10-15:KW_THEN
11-5:KW_BEGIN
12-7:KW_CALL
12-13:TK_IDENT(HANOI)
12-18:SB_LPAR
12-19:TK_IDENT(N)
12-20:SB_MINUS
12-21:TK_NUMBER(1)
12-22:SB_COMMA
12-23:TK_IDENT(S)
12-24:SB_COMMA
12-25:TK_NUMBER(6)
12-26:SB_MINUS
12-27:TK_IDENT(S)
12-28:SB_MINUS
12-29:TK_IDENT(Z)
12-31:SB_RPAR
12-31:SB_SEMICOLON
13-7:TK_IDENT(I)
13-9:SB_ASSIGN

13-10:TK_IDENT(I)
13-11:SB_PLUS
13-12:TK_NUMBER(1)
13-13:SB_SEMICOLON
14-7:KW_CALL
14-13:TK_IDENT(WRITELN)
14-20:SB_SEMICOLON
15-7:KW_CALL
15-13:TK_IDENT(WRITEI)
15-19:SB_LPAR
15-20:TK_IDENT(I)
15-22:SB_RPAR
15-22:SB_SEMICOLON
16-7:KW_CALL
16-13:TK_IDENT(WRITEI)
16-19:SB_LPAR
16-20:TK_IDENT(N)
16-22:SB_RPAR
16-22:SB_SEMICOLON
17-7:KW_CALL
17-13:TK_IDENT(WRITEI)
17-19:SB_LPAR
17-20:TK_IDENT(S)
17-22:SB_RPAR
17-22:SB_SEMICOLON
18-7:KW_CALL
18-13:TK_IDENT(WRITEI)
18-19:SB_LPAR
18-20:TK_IDENT(Z)
18-22:SB_RPAR
18-22:SB_SEMICOLON
19-7:KW_CALL
19-13:TK_IDENT(HANOI)
19-18:SB_LPAR
19-19:TK_IDENT(N)
19-20:SB_MINUS
19-21:TK_NUMBER(1)
19-22:SB_COMMA
19-23:TK_NUMBER(6)
19-24:SB_MINUS
19-25:TK_IDENT(S)
19-26:SB_MINUS

19-27:TK_IDENT(Z)
19-28:SB_COMMA
19-29:TK_IDENT(Z)
19-31:SB_RPAR
20-5:KW_END
21-1:KW_END
21-4:SB_SEMICOLON
23-1:KW_BEGIN
24-3:KW_FOR
24-8:TK_IDENT(N)
24-11:SB_ASSIGN
24-13:TK_NUMBER(1)
24-16:KW_TO
24-20:TK_NUMBER(4)
24-23:KW_DO
25-5:KW_BEGIN
26-7:KW_FOR
26-12:TK_IDENT(I)
26-14:SB_ASSIGN
26-15:TK_NUMBER(1)
26-18:KW_TO
26-22:TK_NUMBER(4)
26-25:KW_DO
27-9:KW_CALL
27-15:TK_IDENT(WRITEC)
27-21:SB_LPAR
27-23:TK_CHAR(' ')
27-26:SB_RPAR
27-26:SB_SEMICOLON
28-7:KW_CALL
28-13:TK_IDENT(READC)
28-18:SB_LPAR
28-19:TK_IDENT(C)
28-21:SB_RPAR
28-21:SB_SEMICOLON
29-7:KW_CALL
29-13:TK_IDENT(WRITEC)
29-19:SB_LPAR
29-20:TK_IDENT(C)
29-22:SB_RPAR
30-5:KW_END
30-8:SB_SEMICOLON

31-3:TK_IDENT(P)
31-5:SB_ASSIGN
31-6:TK_NUMBER(1)
31-7:SB_SEMICOLON
32-3:TK_IDENT(Q)
32-5:SB_ASSIGN
32-6:TK_NUMBER(2)
32-7:SB_SEMICOLON
33-3:KW_FOR
33-8:TK_IDENT(N)
33-10:SB_ASSIGN
33-11:TK_NUMBER(2)
33-14:KW_TO
33-18:TK_NUMBER(4)
33-21:KW_DO
34-5:KW_BEGIN
35-7:TK_IDENT(I)
35-9:SB_ASSIGN
35-10:TK_NUMBER(0)
35-11:SB_SEMICOLON
36-7:KW_CALL
36-13:TK_IDENT(HANOI)
36-18:SB_LPAR
36-19:TK_IDENT(N)
36-20:SB_COMMA
36-21:TK_IDENT(P)
36-22:SB_COMMA
36-23:TK_IDENT(Q)
36-25:SB_RPAR
36-25:SB_SEMICOLON
37-7:KW_CALL
37-13:TK_IDENT(WRITELN)
38-5:KW_END
39-1:KW_END
39-4:SB_PERIOD

5. Các lỗi gặp trong error.h
- error1_endofcomment.kpl:

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of running the `error1_endofcomment.kpl` test. The output shows tokens 1-9 being identified correctly, but token 10 is flagged as an error because it starts with a comment character (*). The code editor shows the Korn shell script used for testing.

```
test > error1_endofcomment.kpl
1 Program CommentError;
2 /* Testing unclosed comment behavior
3 | This comment starts but never ends properly
4 Const
5 x = 5;
6 Begin
7 /* Another comment
8 Write('Hello');
9 End.
```

```
duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/error1_endofcomment.kpl
1-1:KW_PROGRAM
1-9:TK_IDENT(CommentError)
2-1:SB_SEMICOLON
9-5:End of comment expected!
3-5:TK_IDEN
```

- error2_idenntoolong.kpl:

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of running the `error2_idenntoolong.kpl` test. The output shows tokens 1-3 being identified correctly, but token 4 is flagged as an error because its identifier name exceeds the maximum allowed length of 42 characters. The code editor shows the Korn shell script used for testing.

```
test > error2_idenntoolong.kpl
1 Program IdentTest;
2 Begin
3 | verylongIdentifierNameExceedsMax := 42;
4 End.
```

```
duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/error2_idenntoolong.kpl
1-1:KW_PROGRAM
1-9:TK_IDENT(IdentTest)
2-1:SB_SEMICOLON
2-1:KW_BEGIN
3-18:Identification too long!
3-19:Identification too long!
3-20:Identification too long!
3-21:Identification too long!
3-22:Identification too long!
3-23:Identification too long!
3-24:Identification too long!
3-25:Identification too long!
3-26:Identification too long!
3-27:Identification too long!
3-28:Identification too long!
3-29:Identification too long!
3-30:Identification too long!
3-31:Identification too long!
3-32:Identification too long!
3-33:Identification too long!
3-34:Identification too long!
3-35:TK_IDENT(verylongIdent)
3-37:SB_ASSIGN
3-39:TK_NUMBER(42)
4-1:SB_SEMICOLON
4-1:KW_END
4-4:SB_PERIOD
duongcongthuyet@Duongs-MacBook-Pro Scanner %
```

- error3_invalidcharconstant.kpl:

- error3_invalidcharconstant.kpl:

```

1 Program CharConstantErrors;
2 Var
3   c1, c2, c3 : Char;
4 Begin
5   c1 := 'x'; (* Unclosed character literal *)
6   c2 := 'xyz'; (* Multiple characters in literal *)
7   c3 := ''; (* Empty character literal *)
8   c1 := 'A'
9 End.

```

```

duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/error3_invalidcharconstant.kpl
1-1:KW_PROGRAM
1-24:Identification too long!
1-25:Identification too long!
1-26:Identification too long!
1-9:TK_IDENT('charConstantErr')
2-1:SB_SEMICOLON
2-1:KW_VAR
3-3:TK_IDENT('c1')
3-5:SB_ASSIGN
3-6:TK_IDENT('c2')
3-9:SB_COMMA
3-11:TK_IDENT('c3')
3-14:SB_COLON
3-16:KW_CHAR
4-1:TK_SEMICOLON
4-1:KW_BEGIN
5-3:TK_IDENT('c1')
5-7:SB_ASSIGN
5-9:Invalid const char!
5-9:TK_NONE
6-3:TK_IDENT('c2')
6-7:SB_ASSIGN
6-9:Invalid const char!
6-9:TK_NONE
6-12:TK_IDENT('z')
6-13:Invalid const char!
6-13:TK_NONE
7-3:TK_IDENT('c3')
7-7:SB_ASSIGN
7-9:Invalid const char!
7-9:TK_NONE
8-3:TK_IDENT('c1')
8-7:SB_ASSIGN
9-1:Invalid const char!
9-2:TK_NONE
9-1:KW_END
9-4:SB_PERIOD
duongcongthuyet@Duongs-MacBook-Pro Scanner %

```

- error4_invalidsymbol.kpl:

```

1 Program SymbolTest;
2 Var x : Integer;
3 Begin
4   x := 10;
5   @ # $ %
6 End.

```

```

duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/error4_invalidsymbol.kpl
1-1:KW_PROGRAM
1-9:TK_IDENT('SymbolTest')
2-1:SB_SEMICOLON
2-1:KW_VAR
2-3:TK_IDENTIFIER('x')
2-7:SB_COLON
2-9:KW_INTEGER
3-1:SB_SEMICOLON
3-1:KW_BEGIN
4-3:TK_IDENTIFIER('x')
4-4:SB_ASSIGN
4-8:TK_NUMBER('10')
5-1:SB_SEMICOLON
5-3:Invalid symbol!
5-3:TK_NONE
5-5:Invalid symbol!
5-5:TK_NONE
5-7:Invalid symbol!
5-7:TK_NONE
5-9:Invalid symbol!
5-9:TK_NONE
6-1:KW_END
6-4:SB_PERIOD
duongcongthuyet@Duongs-MacBook-Pro Scanner %

```

- error5_numbertoolong.kpl:

The screenshot shows a code editor interface with a dark theme. On the left is the file explorer, displaying a project structure under 'SCANNER'. The current file open in the editor is 'error5_numbertoolong.kpl'. The code in the editor is:

```
test > error5_numbertoolong.kpl
1 Program NumTest;
2 Var x : Integer;
3 Begin
4   x := 99999999999999999999;
5 End.
```

Below the editor is a terminal window showing the output of running the script:

```
duongcongthuyet@Duongs-MacBook-Pro Scanner % ./scanner test/error5_numbertoolong.kpl
1:KW_PROGRAM
1-9:TK_IDENT(NumTest)
2--1:SB_SEMICOLON
2-1:KW_BEGIN
2-5:TK_IDENT(x)
2-7:SB_COLON
2-9:KW_INTEGER
3--1:SB_SEMICOLON
3-1:KW_END
4-4:TK_IDENT(x)
4-6:SB_ASSIGN
4-23:Value of integer number exceeds the range!
4-24:Value of integer number exceeds the range!
4-25:Value of integer number exceeds the range!
4-26:Value of integer number exceeds the range!
4-27:Value of integer number exceeds the range!
4-8:TK_NUMBER(9999999999999999)
5-1:SB_SEMICOLON
5-1:KW_END
5-4:SB_PERIOD
duongcongthuyet@Duongs-MacBook-Pro Scanner %
```

At the bottom of the terminal window, there is a scroll bar indicating the output is scrollable.