

Name: Nguyễn Hữu Hoàng Hải Anh

Student ID: 20226010

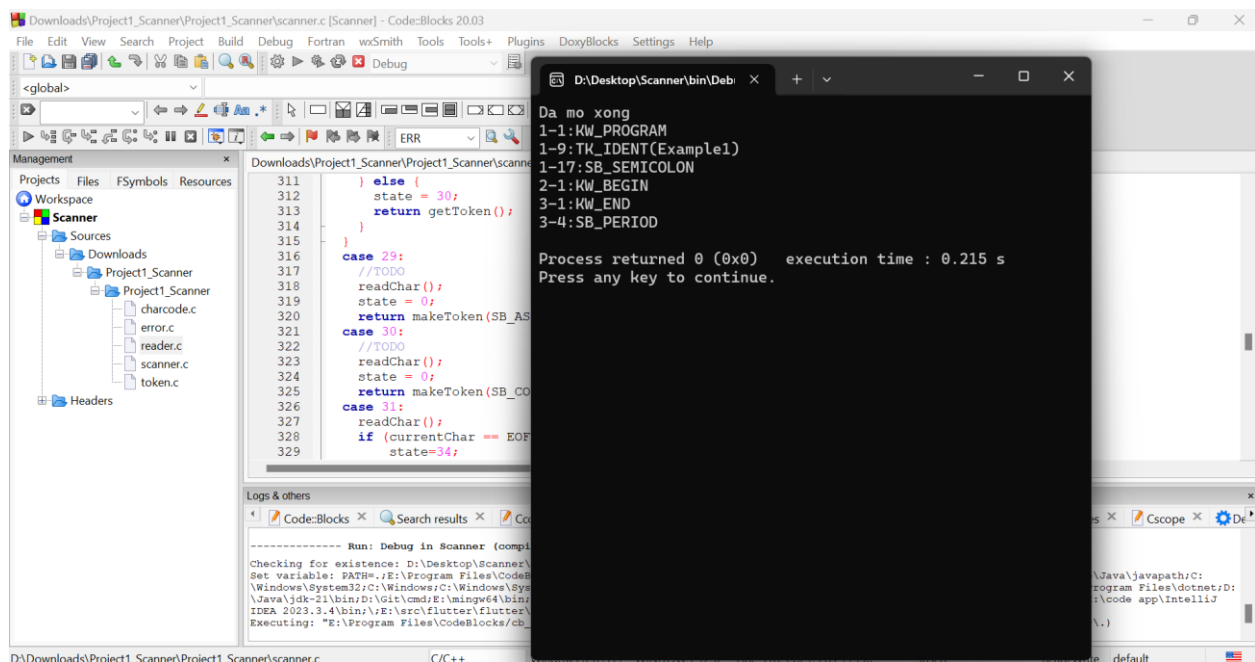
LAB 1. SCANNER

This report includes

- A. Results received when executing project with files
- B. All types of errors that can be found by the scanner and other KPL test files
- C. 43 Cases follow the correct automata

A, Results received when executing project with files

1. Example 1 (*example1.kpl*)



```
Da mo xong
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

Process returned 0 (0x0)   execution time : 0.215 s
Press any key to continue.
```

Results received:

Da mo xong

1-1:KW_PROGRAM

1-9:TK_IDENT(Example1)

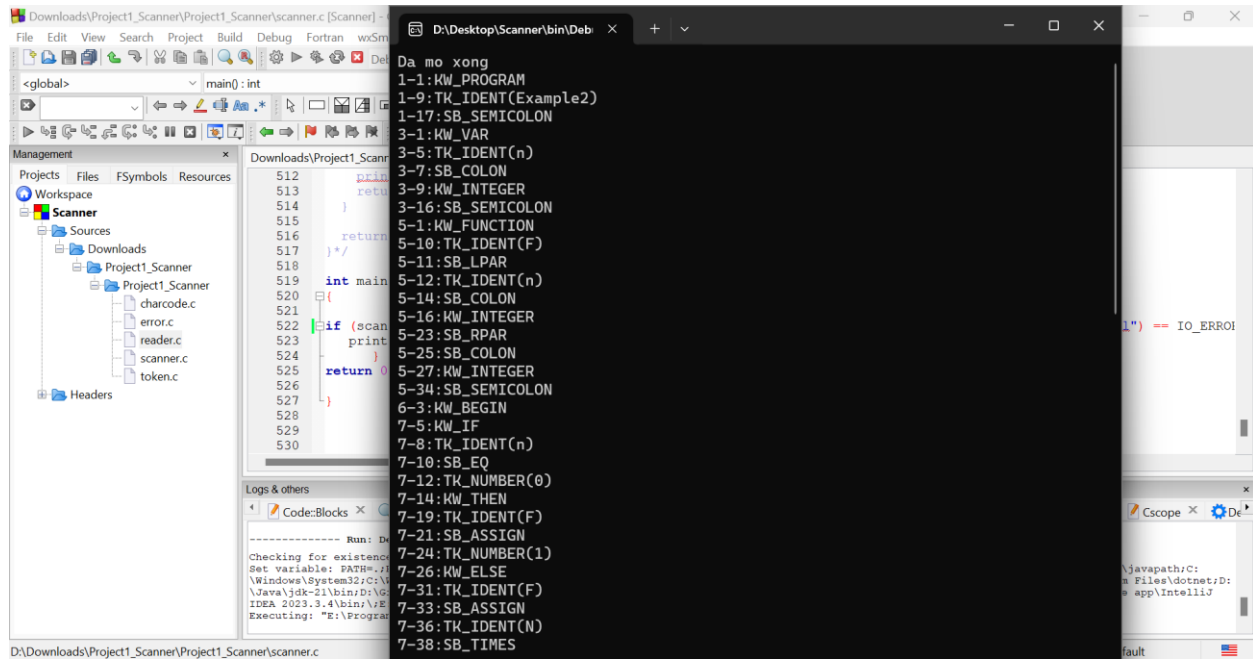
1-17:SB_SEMICOLON

2-1:KW_BEGIN

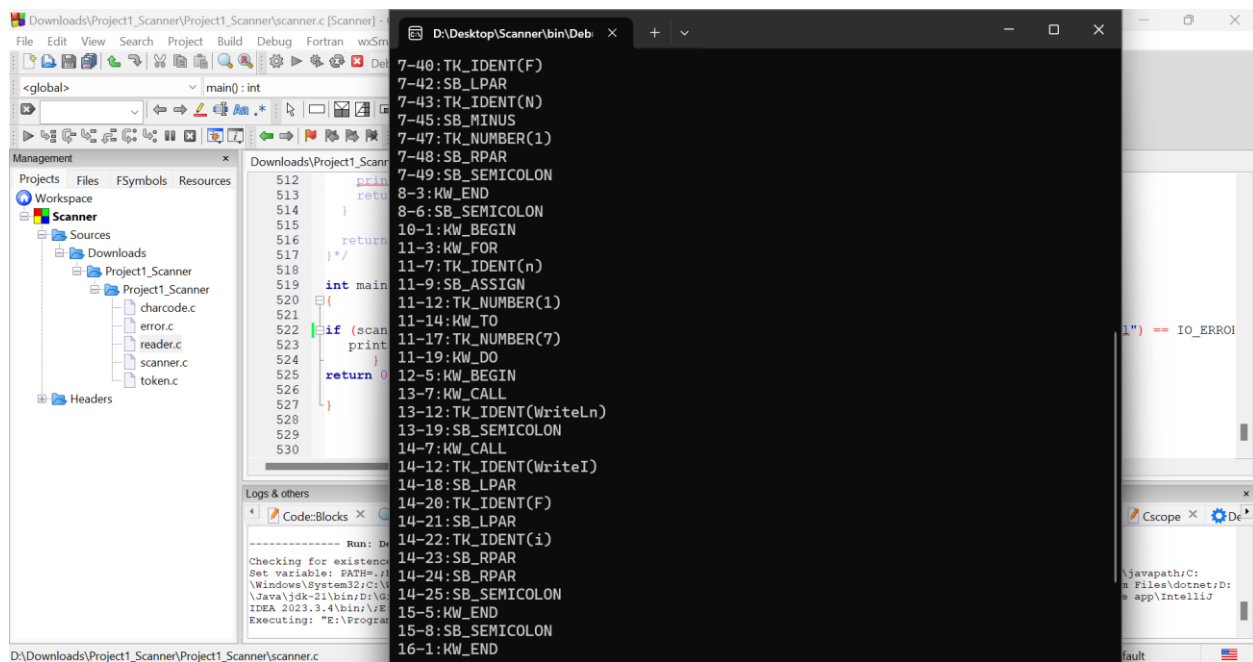
3-1:KW_END

3-4:SB_PERIOD

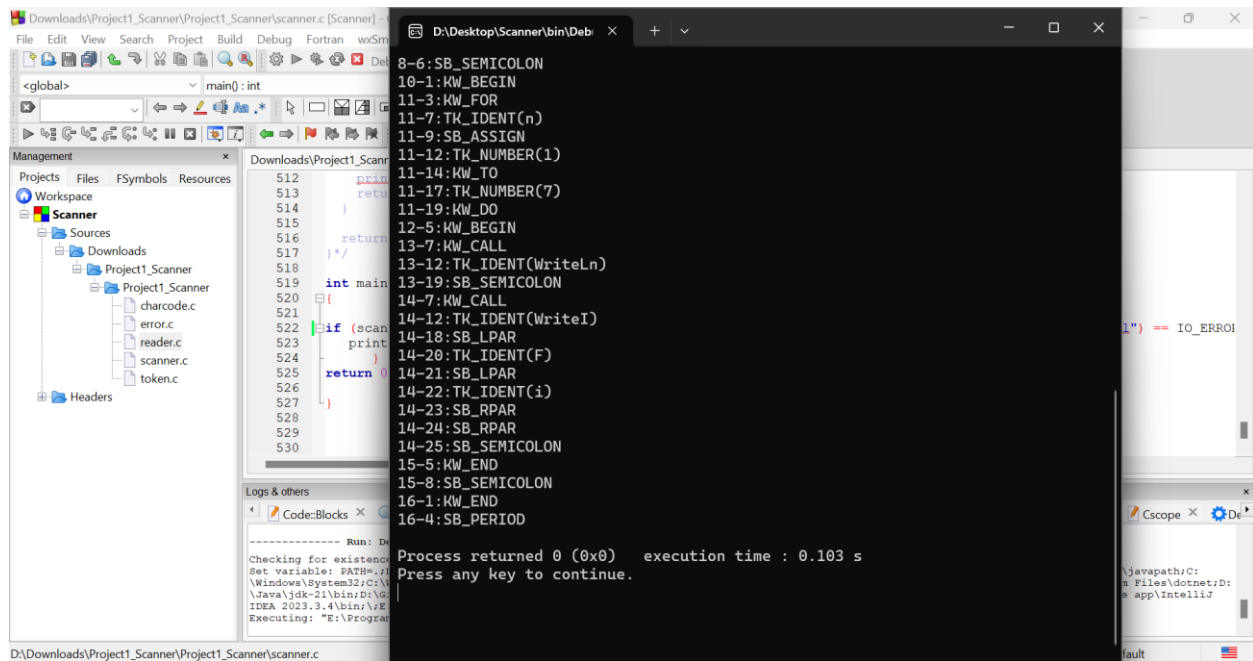
2. Example 2 (*example2.kpl*)



Img1. Line 1 – Line 7 (38)



Img2. Line 7 (40) – Line 16 (1)



Img3. Line 16 (4)

Results received:

Da mo xong

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

3-16: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
5-23:SB_RPAR
5-25:SB_COLON
5-27:KW_INTEGER
5-34: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)
7-21:SB_ASSIGN
7-24:TK_NUMBER(1)
7-26:KW_ELSE
7-31:TK_IDENT(F)
7-33: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)
7-48:SB_RPAR

7-49:SB_SEMICOLON

8-3:KW_END

8-6:SB_SEMICOLON

10-1:KW_BEGIN

11-3:KW_FOR

11-7:TK_IDENT(n)

11-9: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)

13-19: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)

14-23:SB_RPAR

14-24:SB_RPAR

14-25:SB_SEMICOLON

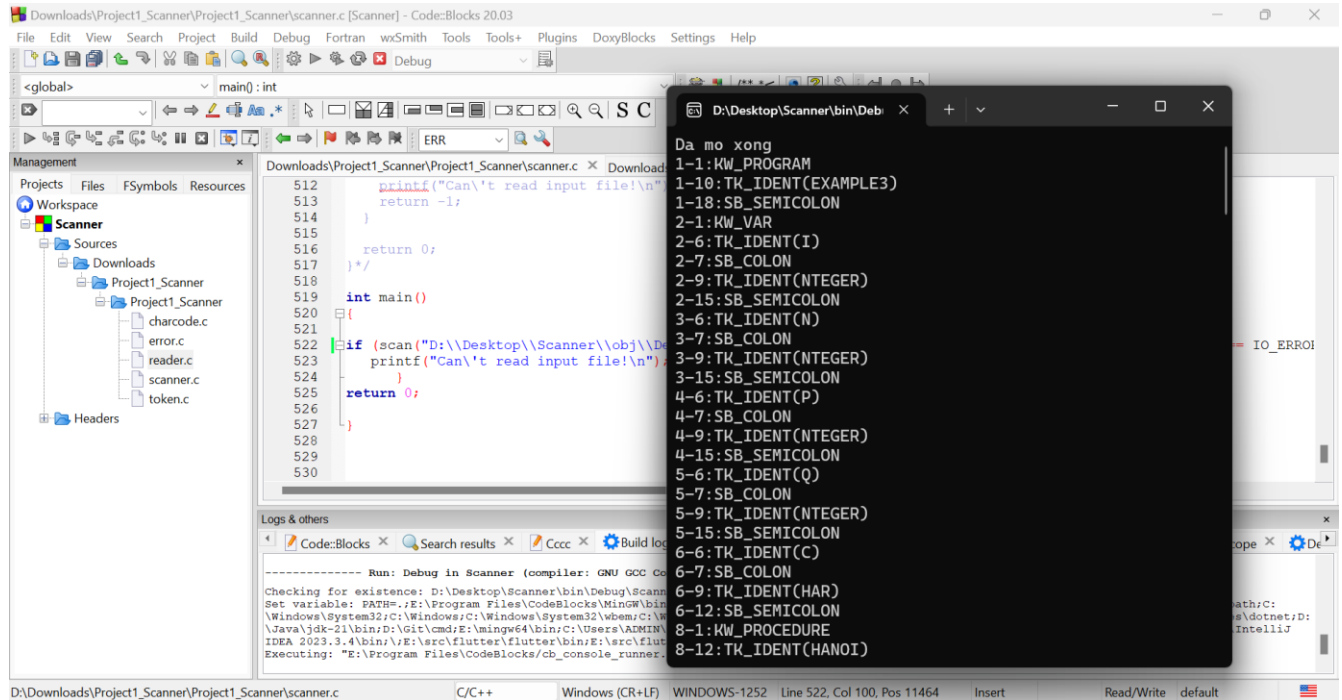
15-5:KW_END

15-8:SB_SEMICOLON

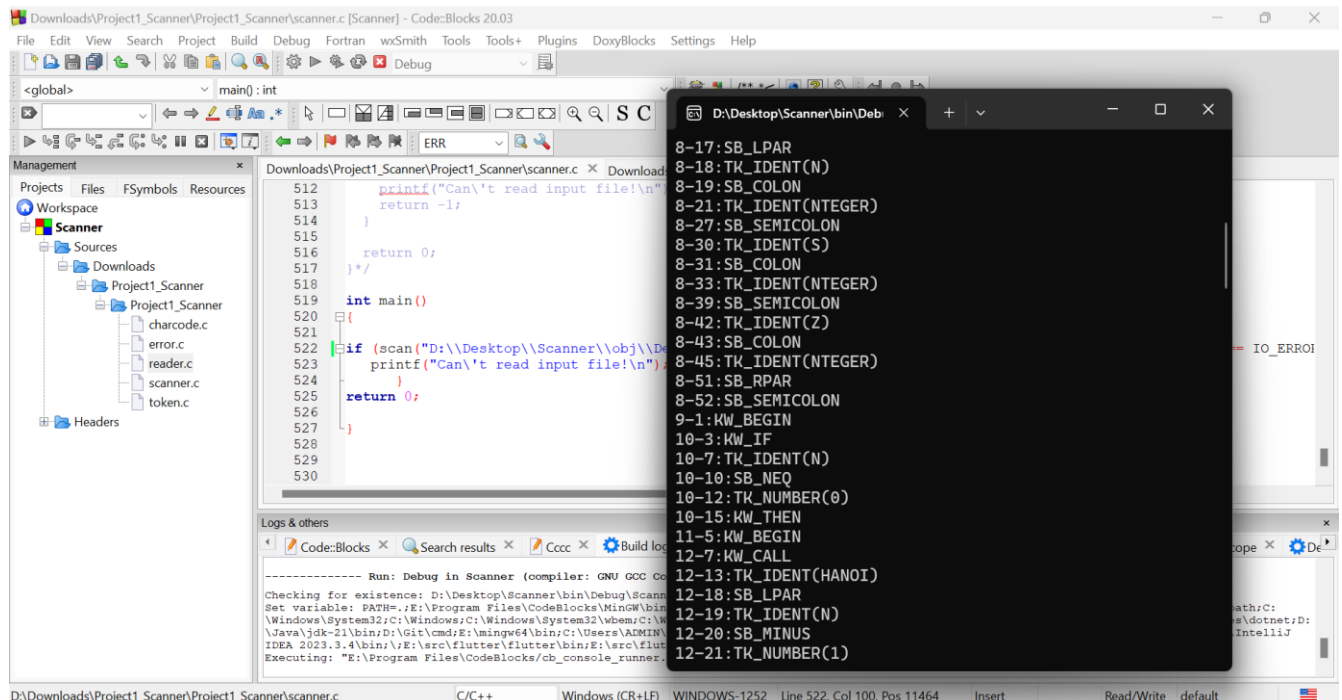
16-1:KW_END

16-4:SB_PERIOD

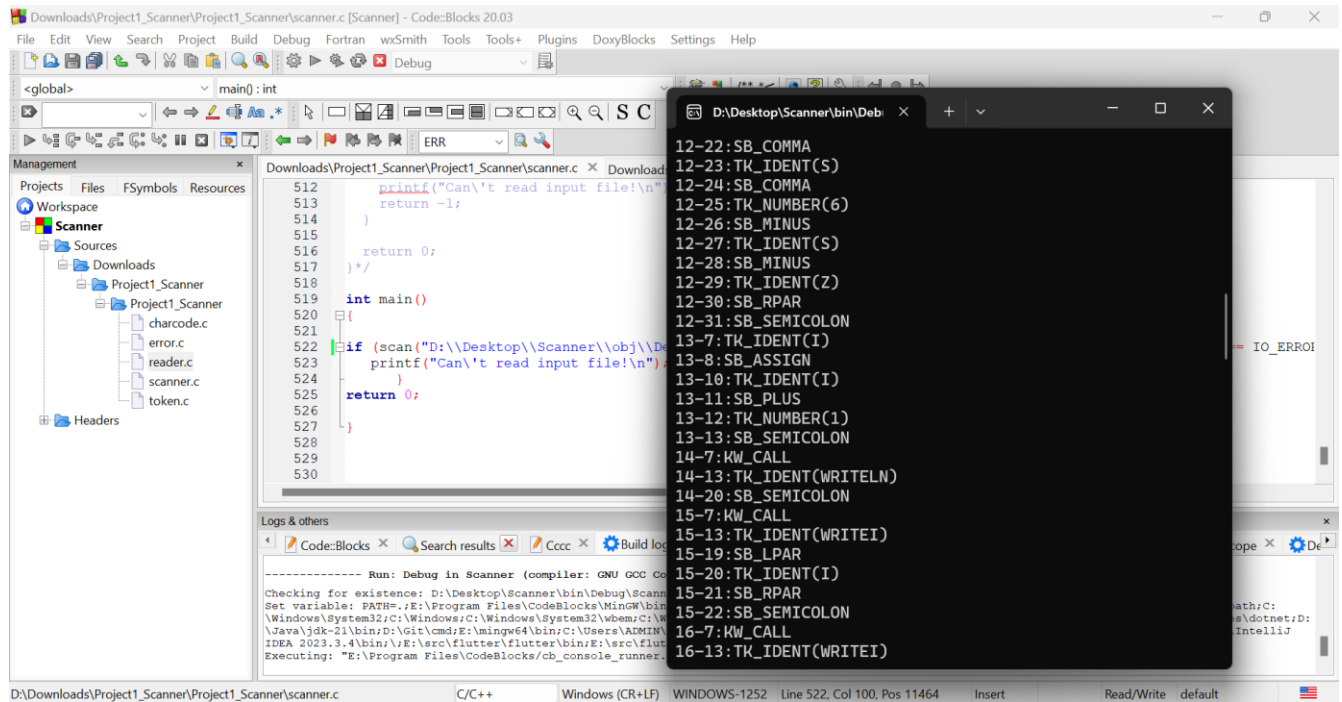
3. Example 3 (*example3.kpl*)



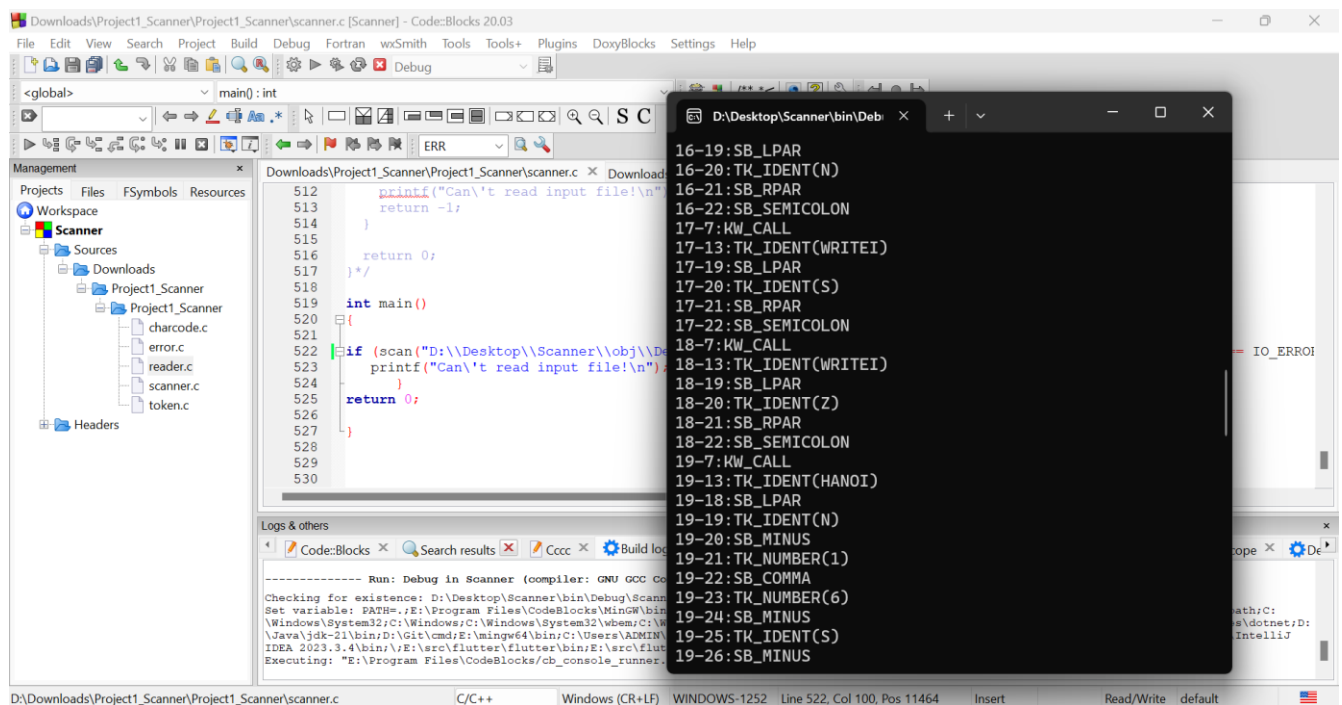
Img4. Line 1 – Line 8 (12)



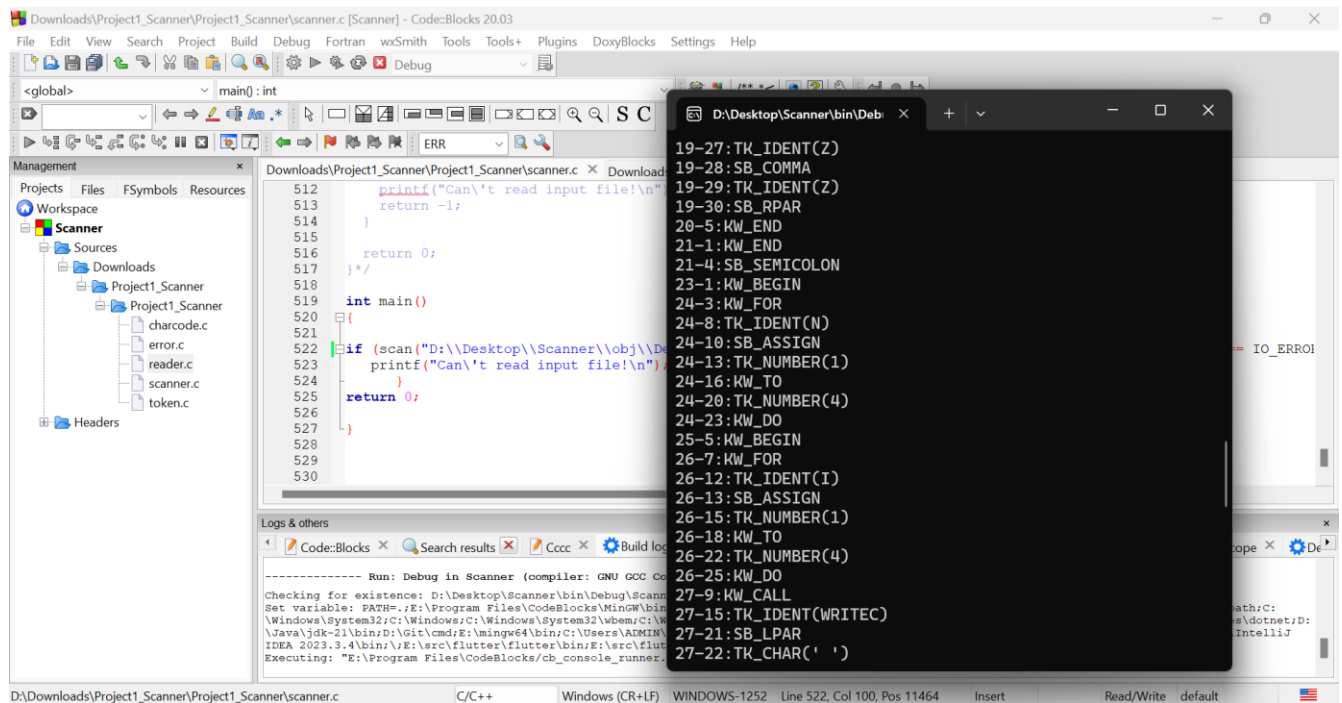
Img5. Line 8 (17) – Line 12 (21)



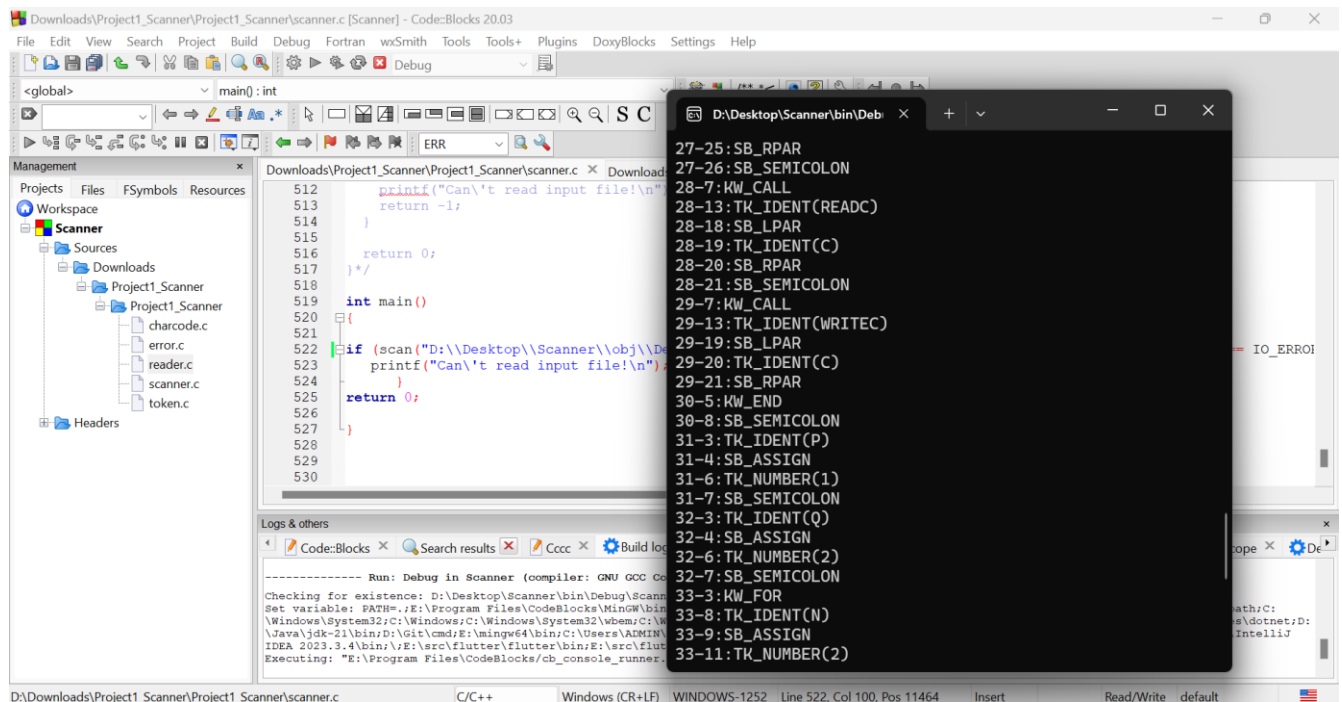
Img6. Line 12 (22) – Line 16 (13)



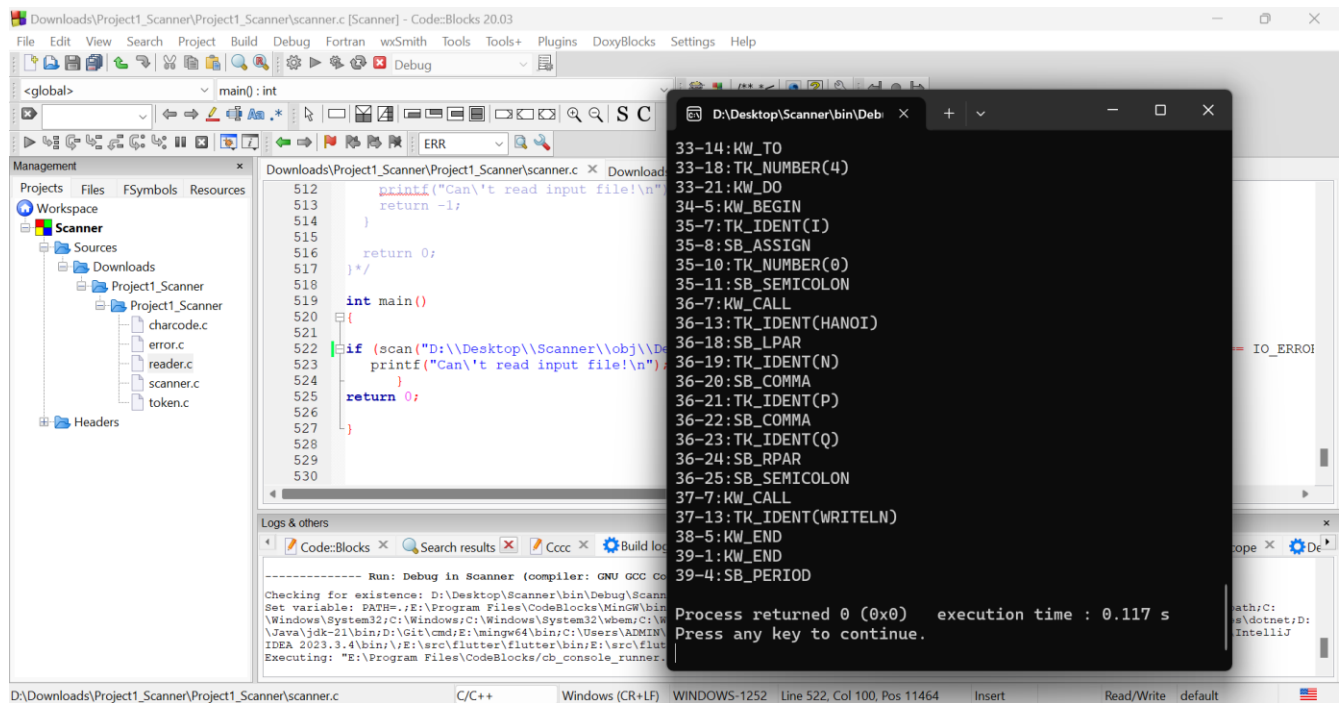
Img7. Line 16 (19) – Line 19 (26)



Img8. Line 19 (27) – Line 27 (22)



Img9. Line 27 (25) – Line 33 (11)



Img10. Line 33 (14) – Line 39

Results received:

Da mo xong

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(Q)
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
6-12: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
8-51:SB_RPAR
8-52:SB_SEMICOLON
9-1:KW_BEGIN
10-3:KW_IF
10-7:TK_IDENT(N)
10-9: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)
12-30:SB_RPAR

12-31:SB_SEMICOLON

13-7:TK_IDENT(I)

13-8: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)

15-21:SB_RPAR

15-22:SB_SEMICOLON

16-7:KW_CALL

16-13:TK_IDENT(WRITEI)

16-19:SB_LPAR

16-20:TK_IDENT(N)

16-21:SB_RPAR

16-22:SB_SEMICOLON

17-7:KW_CALL

17-13:TK_IDENT(WRITEI)

17-19:SB_LPAR

17-20:TK_IDENT(S)

17-21:SB_RPAR

17-22:SB_SEMICOLON

18-7:KW_CALL

18-13:TK_IDENT(WRITEI)

18-19:SB_LPAR

18-20:TK_IDENT(Z)

18-21: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)

19-30: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)

24-10: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)

26-13: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

27-22:TK_CHAR(' ')

27-25:SB_RPAR

27-26:SB_SEMICOLON

28-7:KW_CALL

28-13:TK_IDENT(READC)

28-18:SB_LPAR

28-19:TK_IDENT(C)

28-20:SB_RPAR

28-21:SB_SEMICOLON

29-7:KW_CALL

29-13:TK_IDENT(WRITEC)

29-19:SB_LPAR

29-20:TK_IDENT(C)

29-21:SB_RPAR

30-5:KW_END

30-8:SB_SEMICOLON

31-3:TK_IDENT(P)

31-4:SB_ASSIGN

31-6:TK_NUMBER(1)

31-7:SB_SEMICOLON

32-3:TK_IDENT(Q)

32-4:SB_ASSIGN

32-6:TK_NUMBER(2)

32-7:SB_SEMICOLON

33-3:KW_FOR

33-8:TK_IDENT(N)

33-9: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)
35-8: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)
38-5:KW_END
39-1:KW_END
39-4:SB_PERIOD

B, All types of errors that can be found by the scanner and other KPL test files

Error Handling: Scanner handles 5 unusual cases

- Identifier too long (ERR_IDENTTOOLONG)
- Number too long (ERR_NUMBERTOOLONG)
- Invalid character/symbol (ERR_INVALIDSYMBOL)
- Invalid character constant (ERR_INVALIDCHARCONSTANT)
- Comment not ended (ERR_ENDOFCOMMENT)

I have built a KPL program myself to detect and test all the errors defined in the lab with comments about the corresponding errors.

```
PROGRAM TestErrors;
```

```
VAR
```

```
  ThisIdentTooLong: INTEGER; (* ERR_IDENTTOOLONG *)
```

```
  A: INTEGER;
```

```
  B: INTEGER;
```

```
  C: INTEGER;
```

```
BEGIN
```

```
  A := 12345678901; (* ERR_NUMBERTOOLONG *)
```

```
  B := 2;
```

```
  C := '&&'; (*ERR_INVALIDCHARCONSTANT*)
```

```
  C := @ (*ERR_INVALIDSYMBOL*)
```

```
  (* This is an unterminated comment --> ERR_ENDOFCOMMENT
```

```
END.
```

err1.kpl

12-8:Invalid const char!

12-10:Invalid symbol!

12-11:Invalid const char!

13-8:Invalid symbol!

16-1:End of comment expected!

C, 43 Cases follow the correct automata

I have implemented 43 cases as automata in which I have added some more commands to sample cases as well as completed the //ToDo cases.

Professors and teaching assistants can view the submission in the scanner.c file.

For the sample cases:

- There are cases where I added the line “state = 0;” to continue identifying the rest.
- Store the starting position of each token with two variables ln (row) and cn (column) to help printToken and error know the correct position to print.
- Fix some errors to correct automata

For the //ToDo cases:

- Complete according to automata and add appropriate error detection

Add the necessary declarations

- Example: char numStr[20];

Other parts:

- I added newline characters to the reader.c file to make the printed result look nicer, for example "Da mo xong \n"