Error file syntax and lexical

Source:

program fib(input, output);

var n: integer; var p: integer;

var q: real;

var numsArray : array [13..12] of integer;?

procedure fib(a : integer; b : real; c : real);

begin

if a <= 1 then fib := c

else call fib(a - 1, c, b + c)

end;

procedure fib2(a : integer)

var b : integer; var c : integer; var sum : integer;

procedure rawr3(b : real);

var q : integer;

q := b + 2.0.1;

call fib2(q)

end;

begin

a := aasdlfjlwkjerjkwle - 1;

b := 0;

sum := 1;

claksdjfasdflkaj := b;

while (a\_ > 0) do

begin

a := a - 1;

b := sum;

sum := c + sum;

c := b

end;

fib2 := sum

end;

procedure init;

begin

n := 12;

if (123.4.5 and 2) or 3 then p := 12

else p := 14;

numsArray[3.1] := 15.56;

q := 12

end;

begin

call init;

call rawr3(34);

call writeln(+6\*q/p + 4);

call writeln(fib2\*n);

call writeln(numsArray[3] mod 15)

end.

Listing:

1. program fib(input, output);

2. var n: integer; var p: integer;

3. var q: real;

4. var numsArray : array [13..12] of integer;

5.

6. procedure fib(a : integer; b : real; c : real);

7. begin

8. if a <= 1 then fib := c

9. else call fib(a - 1, c, b + c)

10. end;

11.

12. procedure fib2(a : integer)

13. var b : integer; var c : integer; var sum : integer;

-SYNTAX ERROR-

Expecting one of: ;

Received: var

-SYNTAX ERROR-

Expecting one of: procedure, begin or var

Received: b

-SYNTAX ERROR-

Expecting one of: procedure or begin

Received: var

14. procedure rawr3(b : real);

15. var q : integer;

16.

17. q := b + 2.0;

18. call fib2(q)

19. end;

20. begin

21. a := aasdlfjlwkjerjkwle - 1;

LEXERR: IDTOOLONG aasdlfjlwkjerjkwle

-SYNTAX ERROR-

Expecting one of: ID, num, (, not +, -

Received: aasdlfjlwkjerjkwle

22. b := 0;

23. sum := 1;

24. c := b;

25. while (a\_\_\_\_\_\_\_\_ > 0) do

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

LEXERR: UnrecognizedSymbol \_

-SYNTAX ERROR-

Expecting one of: [ mulop addop relop do then ] , ) ; end else

Received: \_

26. begin

27. a := a - 1;

28. b := sum;

29. sum := c + sum;

30. c := b

31. end;

32. fib2 := sum

33. end;

-SYNTAX ERROR-

Expecting one of: .

Received: ;

34.

35. procedure init;

-SYNTAX ERROR-

Expecting one of: EOF

Received: procedure

Tokens:

Line No. Lexeme Token Type Attribute

1 program 30 0

1 fib 1 0x7ffbd1c02b00

1 ( 2 81

1 input 1 0x7ffbd1c02b80

1 , 4 85

1 output 1 0x7ffbd1c02c00

1 ) 2 82

1 ; 4 86

2 var 31 0

2 n 1 0x7ffbd1c02ce0

2 : 6 0

2 integer 34 0

2 ; 4 86

2 var 31 0

2 p 1 0x7ffbd1c02df0

2 : 6 0

2 integer 34 0

2 ; 4 86

3 var 31 0

3 q 1 0x7ffbd1c02f00

3 : 6 0

3 real 35 0

3 ; 4 86

4 var 31 0

4 numsArray 1 0x7ffbd1c03010

4 : 6 0

4 array 32 0

4 [ 2 83

4 13 10 0

4 .. 5 0

4 12 10 0

4 ] 2 84

4 of 33 0

4 integer 34 0

4 ; 4 86

6 procedure 37 0

6 fib 1 0x7ffbd1c02b00

6 ( 2 81

6 a 1 0x7ffbd1c032d0

6 : 6 0

6 integer 34 0

6 ; 4 86

6 b 1 0x7ffbd1c033b0

6 : 6 0

6 real 35 0

6 ; 4 86

6 c 1 0x7ffbd1c03490

6 : 6 0

6 real 35 0

6 ) 2 82

6 ; 4 86

7 begin 38 70

8 if 39 72

8 a 1 0x7ffbd1c032d0

8 <= 7 88

8 1 10 0

8 then 39 73

8 fib 1 0x7ffbd1c02b00

8 := 3 0

8 c 1 0x7ffbd1c03490

9 else 39 74

9 call 43 0

9 fib 1 0x7ffbd1c02b00

9 ( 2 81

9 a 1 0x7ffbd1c032d0

9 - 9 97

9 1 10 0

9 , 4 85

9 c 1 0x7ffbd1c03490

9 , 4 85

9 b 1 0x7ffbd1c033b0

9 + 9 96

9 c 1 0x7ffbd1c03490

9 ) 2 82

10 end 38 71

10 ; 4 86

12 procedure 37 0

12 fib2 1 0x7ffbd1c03a80

12 ( 2 81

12 a 1 0x7ffbd1c032d0

12 : 6 0

12 integer 34 0

12 ) 2 82

13 var 31 0

13 b 1 0x7ffbd1c033b0

13 : 6 0

13 integer 34 0

13 ; 4 86

13 var 31 0

13 c 1 0x7ffbd1c03490

13 : 6 0

13 integer 34 0

13 ; 4 86

13 var 31 0

13 sum 1 0x7ffbd1c03dd0

13 : 6 0

13 integer 34 0

13 ; 4 86

14 procedure 37 0

14 rawr3 1 0x7ffbd1c03ee0

14 ( 2 81

14 b 1 0x7ffbd1c033b0

14 : 6 0

14 real 35 0

14 ) 2 82

14 ; 4 86

15 var 31 0

15 q 1 0x7ffbd1c02f00

15 : 6 0

15 integer 34 0

15 ; 4 86

17 q 1 0x7ffbd1c02f00

17 := 3 0

17 b 1 0x7ffbd1c033b0

17 + 9 96

17 2.0 11 0

17 ; 4 86

18 call 43 0

18 fib2 1 0x7ffbd1c03a80

18 ( 2 81

18 q 1 0x7ffbd1c02f00

18 ) 2 82

19 end 38 71

19 ; 4 86

20 begin 38 70

21 a 1 0x7ffbd1c032d0

21 := 3 0

21 aasdlfjlwkjerjkwle 99 100

21 - 9 97

21 1 10 0

21 ; 4 86

22 b 1 0x7ffbd1c033b0

22 := 3 0

22 0 10 0

22 ; 4 86

23 sum 1 0x7ffbd1c03dd0

23 := 3 0

23 1 10 0

23 ; 4 86

24 c 1 0x7ffbd1c03490

24 := 3 0

24 b 1 0x7ffbd1c033b0

24 ; 4 86

25 while 40 75

25 ( 2 81

25 a 1 0x7ffbd1c032d0

25 \_ 99 101

25 \_ 99 101

25 \_ 99 101

25 \_ 99 101

25 \_ 99 101

25 \_ 99 101

25 \_ 99 101

25 \_ 99 101

25 > 7 93

25 0 10 0

25 ) 2 82

25 do 40 76

26 begin 38 70

27 a 1 0x7ffbd1c032d0

27 := 3 0

27 a 1 0x7ffbd1c032d0

27 - 9 97

27 1 10 0

27 ; 4 86

28 b 1 0x7ffbd1c033b0

28 := 3 0

28 sum 1 0x7ffbd1c03dd0

28 ; 4 86

29 sum 1 0x7ffbd1c03dd0

29 := 3 0

29 c 1 0x7ffbd1c03490

29 + 9 96

29 sum 1 0x7ffbd1c03dd0

29 ; 4 86

30 c 1 0x7ffbd1c03490

30 := 3 0

30 b 1 0x7ffbd1c033b0

31 end 38 71

31 ; 4 86

32 fib2 1 0x7ffbd1c03a80

32 := 3 0

32 sum 1 0x7ffbd1c03dd0

33 end 38 71

33 ; 4 86

35 procedure 37 0

35 init 1 0x7ffbd1c04f60

35 ; 4 86

Error file with multiple error types

Source:

program fib(input, output);

var n: integer; var p: integer;

var q: real;

var numsArray : array [13..12] of integer;

procedure fib(a : integer; b : real; c : real);

begin

if a <= 1 then fib := c

else fib(a - 1, c, b + c)

end;

procedure fib2(a : integer);

var b : integer; var c : integer; var sum : integer;

procedure meow3(b : real);

var q : integer;

begin

q := b + 2.0;

call fib2(q)

end;

begin

a := a 1;

b := 0;

sum := 1;

c := b;

while (a > 0) do

begin

a := a - 1;

b := sum;

sum := c + sum;

c := b

;

fib2 := sum

end;

procedure init;

begin

n := 12;

if (1 and 2) or 3 then p := 12

else p := 14

numsArray[3] := 15.56;

q := 12

end;

begin

call init;

call (34);

call writeln(+6\*q/p + 4);

call writeln(fib2\*n);

call writeln(numsArray[3] mod 15)

end.

Listing:

1. program fib(input, output);

2. var n: integer; var p: integer;

3. var q: real;

4. var numsArray : array [13..12] of integer;

5.

6. procedure fib(a : integer; b : real; c : real);

7. begin

8. if a <= 1 then fib := c

9. else fib(a - 1, c, b + c)

-SYNTAX ERROR-

Expecting one of: [, :=

Received: (

10. end;

11.

12. procedure fib2(a : integer);

13. var b : integer; var c : integer; var sum : integer;

14. procedure meow3(b : real);

15. var q : integer;

16. begin

17. q := b + 2.0;

18. call fib2(q)

19. end;

20. begin

21. a := a 1;

-SYNTAX ERROR-

Expecting one of: [ mulop addop relop do then ] , ) ; end else

Received: 1

22. b := 0;

23. sum := 1;

24. c := b;

25. while (a > 0) do

26. begin

27. a := a - 1;

28. b := sum;

29. sum := c + sum;

30. c := b

31. ;

32. fib2 := sum

33. end;

34.

35. procedure init;

-SYNTAX ERROR-

Expecting one of: ID, call, begin, while, if

Received: procedure

36. begin

37. n := 12;

38. if (1 and 2) or 3 then p := 12

39. else p := 14

40. numsArray[3] := 15.56;

-SYNTAX ERROR-

Expecting one of: mulop addop relop do then ] , ) ; end else

Received: numsArray

-SYNTAX ERROR-

Expecting one of: ; end

Received: ]

41. q := 12

42. end;

43.

44. begin

45. call init;

46. call (34);

-SYNTAX ERROR-

Expecting one of: ID

Received: (

-SYNTAX ERROR-

Expecting one of: (, ;, end, else

Received: 34

47. call writeln(+6\*q/p + 4);

48. call writeln(fib2\*n);

49. call writeln(numsArray[3] mod 15)

50. end.

-SYNTAX ERROR-

Expecting one of: ; end

Received: .

-SYNTAX ERROR-

Expecting one of: end

Received: EOF

-SYNTAX ERROR-

Expecting one of: .

Received: EOF

Tokens:

Line No. Lexeme Token Type Attribute

1 program 30 0

1 fib 1 0x7fa7ebc02b00

1 ( 2 81

1 input 1 0x7fa7ebc02b80

1 , 4 85

1 output 1 0x7fa7ebc02c00

1 ) 2 82

1 ; 4 86

2 var 31 0

2 n 1 0x7fa7ebc02ce0

2 : 6 0

2 integer 34 0

2 ; 4 86

2 var 31 0

2 p 1 0x7fa7ebc02df0

2 : 6 0

2 integer 34 0

2 ; 4 86

3 var 31 0

3 q 1 0x7fa7ebc02f00

3 : 6 0

3 real 35 0

3 ; 4 86

4 var 31 0

4 numsArray 1 0x7fa7ebc03010

4 : 6 0

4 array 32 0

4 [ 2 83

4 13 10 0

4 .. 5 0

4 12 10 0

4 ] 2 84

4 of 33 0

4 integer 34 0

4 ; 4 86

6 procedure 37 0

6 fib 1 0x7fa7ebc02b00

6 ( 2 81

6 a 1 0x7fa7ebc032d0

6 : 6 0

6 integer 34 0

6 ; 4 86

6 b 1 0x7fa7ebc033b0

6 : 6 0

6 real 35 0

6 ; 4 86

6 c 1 0x7fa7ebc03490

6 : 6 0

6 real 35 0

6 ) 2 82

6 ; 4 86

7 begin 38 70

8 if 39 72

8 a 1 0x7fa7ebc032d0

8 <= 7 88

8 1 10 0

8 then 39 73

8 fib 1 0x7fa7ebc02b00

8 := 3 0

8 c 1 0x7fa7ebc03490

9 else 39 74

9 fib 1 0x7fa7ebc02b00

9 ( 2 81

9 a 1 0x7fa7ebc032d0

9 - 9 97

9 1 10 0

9 , 4 85

9 c 1 0x7fa7ebc03490

9 , 4 85

9 b 1 0x7fa7ebc033b0

9 + 9 96

9 c 1 0x7fa7ebc03490

9 ) 2 82

10 end 38 71

10 ; 4 86

12 procedure 37 0

12 fib2 1 0x7fa7ebc03a50

12 ( 2 81

12 a 1 0x7fa7ebc032d0

12 : 6 0

12 integer 34 0

12 ) 2 82

12 ; 4 86

13 var 31 0

13 b 1 0x7fa7ebc033b0

13 : 6 0

13 integer 34 0

13 ; 4 86

13 var 31 0

13 c 1 0x7fa7ebc03490

13 : 6 0

13 integer 34 0

13 ; 4 86

13 var 31 0

13 sum 1 0x7fa7ebc03dd0

13 : 6 0

13 integer 34 0

13 ; 4 86

14 procedure 37 0

14 meow3 1 0x7fa7ebc03ee0

14 ( 2 81

14 b 1 0x7fa7ebc033b0

14 : 6 0

14 real 35 0

14 ) 2 82

14 ; 4 86

15 var 31 0

15 q 1 0x7fa7ebc02f00

15 : 6 0

15 integer 34 0

15 ; 4 86

16 begin 38 70

17 q 1 0x7fa7ebc02f00

17 := 3 0

17 b 1 0x7fa7ebc033b0

17 + 9 96

17 2.0 11 0

17 ; 4 86

18 call 43 0

18 fib2 1 0x7fa7ebc03a50

18 ( 2 81

18 q 1 0x7fa7ebc02f00

18 ) 2 82

19 end 38 71

19 ; 4 86

20 begin 38 70

21 a 1 0x7fa7ebc032d0

21 := 3 0

21 a 1 0x7fa7ebc032d0

21 1 10 0

21 ; 4 86

22 b 1 0x7fa7ebc033b0

22 := 3 0

22 0 10 0

22 ; 4 86

23 sum 1 0x7fa7ebc03dd0

23 := 3 0

23 1 10 0

23 ; 4 86

24 c 1 0x7fa7ebc03490

24 := 3 0

24 b 1 0x7fa7ebc033b0

24 ; 4 86

25 while 40 75

25 ( 2 81

25 a 1 0x7fa7ebc032d0

25 > 7 93

25 0 10 0

25 ) 2 82

25 do 40 76

26 begin 38 70

27 a 1 0x7fa7ebc032d0

27 := 3 0

27 a 1 0x7fa7ebc032d0

27 - 9 97

27 1 10 0

27 ; 4 86

28 b 1 0x7fa7ebc033b0

28 := 3 0

28 sum 1 0x7fa7ebc03dd0

28 ; 4 86

29 sum 1 0x7fa7ebc03dd0

29 := 3 0

29 c 1 0x7fa7ebc03490

29 + 9 96

29 sum 1 0x7fa7ebc03dd0

29 ; 4 86

30 c 1 0x7fa7ebc03490

30 := 3 0

30 b 1 0x7fa7ebc033b0

31 ; 4 86

32 fib2 1 0x7fa7ebc03a50

32 := 3 0

32 sum 1 0x7fa7ebc03dd0

33 end 38 71

33 ; 4 86

35 procedure 37 0

35 init 1 0x7fa7ebc049e0

35 ; 4 86

36 begin 38 70

37 n 1 0x7fa7ebc02ce0

37 := 3 0

37 12 10 0

37 ; 4 86

38 if 39 72

38 ( 2 81

38 1 10 0

38 and 8 80

38 2 10 0

38 ) 2 82

38 or 9 77

38 3 10 0

38 then 39 73

38 p 1 0x7fa7ebc02df0

38 := 3 0

38 12 10 0

39 else 39 74

39 p 1 0x7fa7ebc02df0

39 := 3 0

39 14 10 0

40 numsArray 1 0x7fa7ebc03010

40 [ 2 83

40 3 10 0

40 ] 2 84

40 := 3 0

40 15.56 11 0

40 ; 4 86

41 q 1 0x7fa7ebc02f00

41 := 3 0

41 12 10 0

42 end 38 71

42 ; 4 86

44 begin 38 70

45 call 43 0

45 init 1 0x7fa7ebc049e0

45 ; 4 86

46 call 43 0

46 ( 2 81

46 34 10 0

46 ) 2 82

46 ; 4 86

47 call 43 0

47 writeln 1 0x7fa7ebc05270

47 ( 2 81

47 + 9 96

47 6 10 0

47 \* 8 94

47 q 1 0x7fa7ebc02f00

47 / 8 95

47 p 1 0x7fa7ebc02df0

47 + 9 96

47 4 10 0

47 ) 2 82

47 ; 4 86

48 call 43 0

48 writeln 1 0x7fa7ebc05270

48 ( 2 81

48 fib2 1 0x7fa7ebc03a50

48 \* 8 94

48 n 1 0x7fa7ebc02ce0

48 ) 2 82

48 ; 4 86

49 call 43 0

49 writeln 1 0x7fa7ebc05270

49 ( 2 81

49 numsArray 1 0x7fa7ebc03010

49 [ 2 83

49 3 10 0

49 ] 2 84

49 mod 8 79

49 15 10 0

49 ) 2 82

50 end 38 71

50 . 4 87

-1 EOF 20 0