**Test Input 1**

with std.io;

with gnu.io;

@@

A : Integer;

B : float = 3.42;

function foo(X:Integer;) return integer

{

--Function Code Goes in here

get(X);

X=X+2;

return X;

}

@@

A=foo(A);

put(A);

put(B);

**Output 1**

1 with Keyword

1 std.io Keyword

1 ; Delimiter

2 with Keyword

2 gnu.io Keyword

2 ; Delimiter

4 @@ Section\_Separator

6 A Identifier

6 : Delimiter

6 Integer Keyword

6 ; Delimiter

7 B Identifier

7 : Delimiter

7 float Keyword

7 = Operator

7 3.42 Floating\_Point\_Literal

7 ; Delimiter

9 function Keyword

9 foo Identifier

9 ( Delimiter

9 X Identifier

9 : Delimiter

9 Integer Keyword

9 ; Delimiter

9 ) Delimiter

9 return Keyword

9 integer Keyword

10 { Delimiter

12 get Keyword

12 ( Delimiter

12 X Identifier

12 ) Delimiter

12 ; Delimiter

13 X Identifier

13 = Operator

13 X Identifier

13 + Operator

13 2 Integer\_Literal

13 ; Delimiter

15 return Keyword

15 X Identifier

15 ; Delimiter

16 } Delimiter

18 @@ Section\_Separator

20 A Identifier

20 = Operator

20 foo Identifier

20 ( Delimiter

20 A Identifier

20 ) Delimiter

20 ; Delimiter

21 put Keyword

21 ( Delimiter

21 A Identifier

21 ) Delimiter

21 ; Delimiter

22 put Keyword

22 ( Delimiter

22 B Identifier

22 ) Delimiter

22 ; Delimiter

**Test Input 2**

With std.io;

wITh gnu.io;

@@

A : InteGer;

B : aRRay[3] of fLoAt;

function foo(X:Integer;) return integer

{

get(X);--comments

X=X+2;

while(variable>=100)

{

if(true)

{}

else

{}

}

return X;

}

@@

/\*multi

line

comments\*/

A=foo(A);

put(A);

put(B[2]);

**Output 2**

1 With Keyword

1 std.io Keyword

1 ; Delimiter

2 wITh Keyword

2 gnu.io Keyword

2 ; Delimiter

4 @@ Section\_Separator

6 A Identifier

6 : Delimiter

6 InteGer Keyword

6 ; Delimiter

7 B Identifier

7 : Delimiter

7 aRRay Keyword

7 [ Delimiter

7 3 Integer\_Literal

7 ] Delimiter

7 of Keyword

7 fLoAt Keyword

7 ; Delimiter

9 function Keyword

9 foo Identifier

9 ( Delimiter

9 X Identifier

9 : Delimiter

9 Integer Keyword

9 ; Delimiter

9 ) Delimiter

9 return Keyword

9 integer Keyword

10 { Delimiter

11 get Keyword

11 ( Delimiter

11 X Identifier

11 ) Delimiter

11 ; Delimiter

12 X Identifier

12 = Operator

12 X Identifier

12 + Operator

12 2 Integer\_Literal

12 ; Delimiter

13 while Keyword

13 ( Delimiter

13 variable Identifier

13 >= Operator

13 100 Integer\_Literal

13 ) Delimiter

14 { Delimiter

15 if Keyword

15 ( Delimiter

15 true Keyword

15 ) Delimiter

16 { Delimiter

16 } Delimiter

17 else Keyword

18 { Delimiter

18 } Delimiter

19 } Delimiter

20 return Keyword

20 X Identifier

20 ; Delimiter

21 } Delimiter

23 @@ Section\_Separator

28 A Identifier

28 = Operator

28 foo Identifier

28 ( Delimiter

28 A Identifier

28 ) Delimiter

28 ; Delimiter

29 put Keyword

29 ( Delimiter

29 A Identifier

29 ) Delimiter

29 ; Delimiter

30 put Keyword

30 ( Delimiter

30 B Identifier

30 [ Delimiter

30 2 Integer\_Literal

30 ] Delimiter

30 ) Delimiter

30 ; Delimiter

**Test Input 3**

@@

"abc123xyz"

5426

1.234

+

&&

==

abc123

;

funCTion

123.abc

1..234

@

9abcxyz

**Output 3**

1 @@ Section\_Separator

2 "abc123xyz" String\_Literal

3 5426 Integer\_Literal

4 1.234 Floating\_Point\_Literal

5 + Operator

6 && Operator

7 == Operator

8 abc123 Identifier

9 ; Delimiter

10 funCTion Keyword

Error at line number 11, Found an unidentified token 123.abc

Error at line number 12, Found an unidentified token 1..234c

Error at line number 13, Found an unidentified token @

Error at line number 14, Found an unidentified token 9abcxyz