Ain Shams University
Faculty of Engineering
Computers and Systems Engineering Department

Project 2 (Systems Software CSE422)

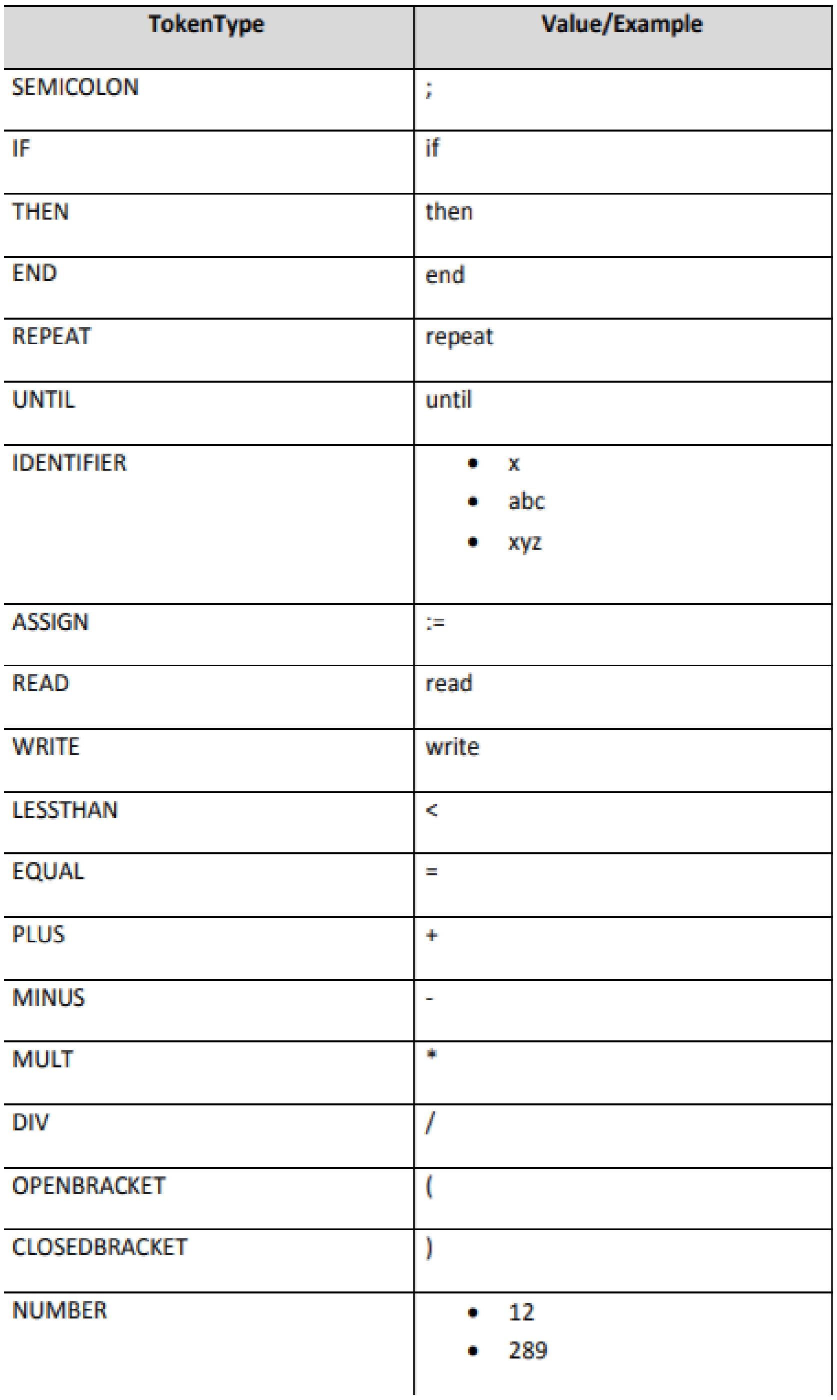
Parser Project

section	code	name
2	1700684	شمس الدين عبدالرحمن علي علي
3	1701041	لؤي باسم نبيه عبد الواحد
3	1701008	كريم مجدي السيد حافظ محمد سليمان
4	1701136	محمد احمد ماهر احمد عبدالرحيم
5	1701730	يوسف عباس احمد عباس
5	1701671	ياسر احمد محمد احمد
5	1701665	وليد عبد العاطي حسن
5	1701588	نور الدين محمود مصباح كمال الحوت

Parser Project

We used C# to build a GUI used to load any text file written by TINY language snippet code and then scanner it and convert grammar into EBNF form then parser it to generate a complete syntax tree.

Tokens of the TINY language



TokenType	Value/Example
SEMICOLON	;
IF	if
THEN	then
END	end
REPEAT	repeat
UNTIL	until
IDENTIFIER	• x • abc • xyz
ASSIGN	:=
READ	read
WRITE	write
LESSTHAN	<
EQUAL	=
PLUS	+
MINUS	-
MULT	*
DIV	/
OPENBRACKET	(
CLOSEDBRACKET	)
NUMBER	• 12 • 289

We open Scanner.exe

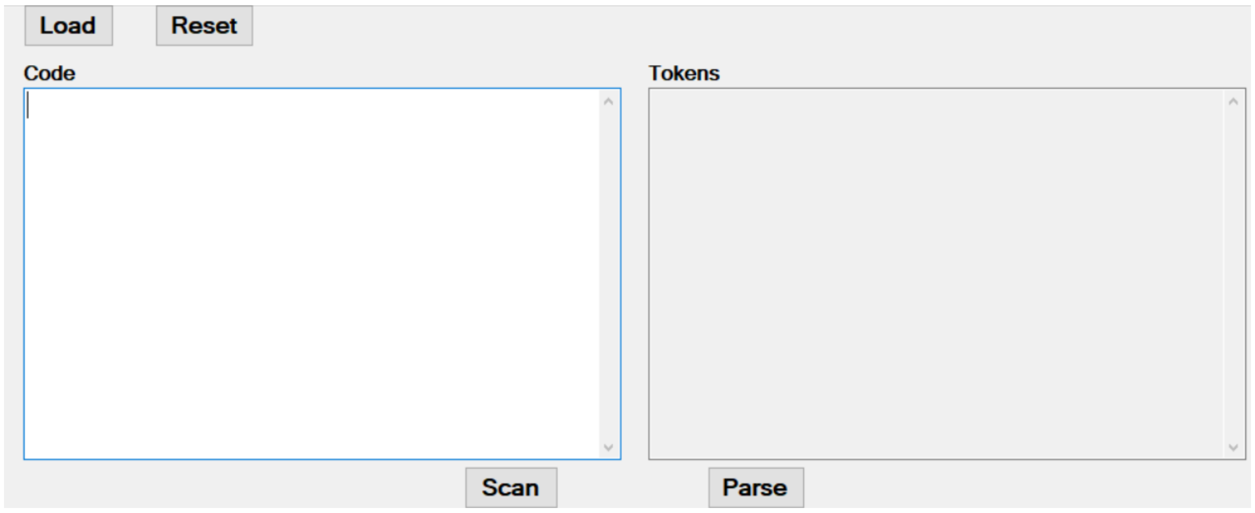
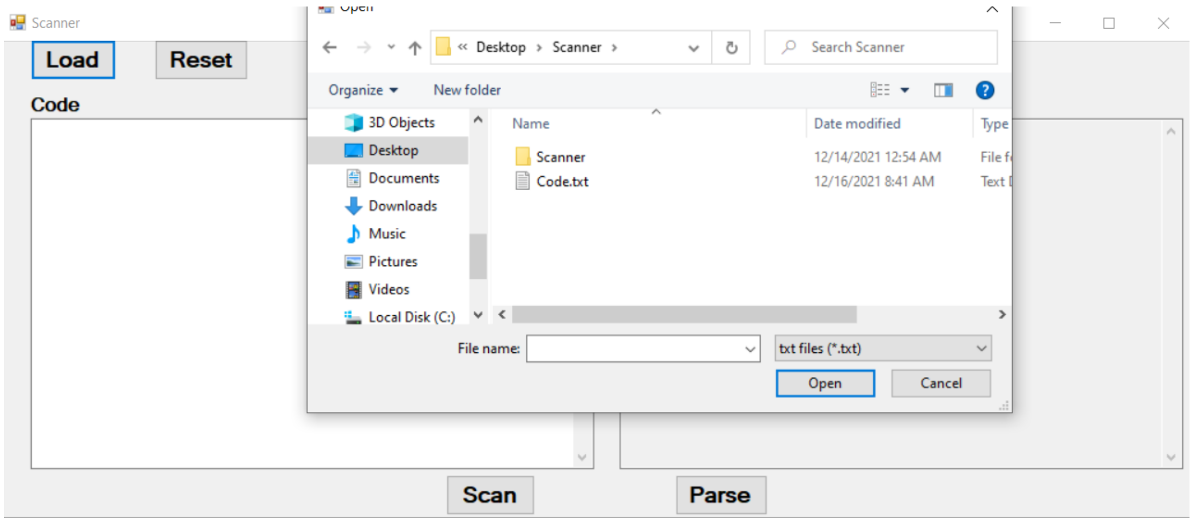


Figure 1 scanner application

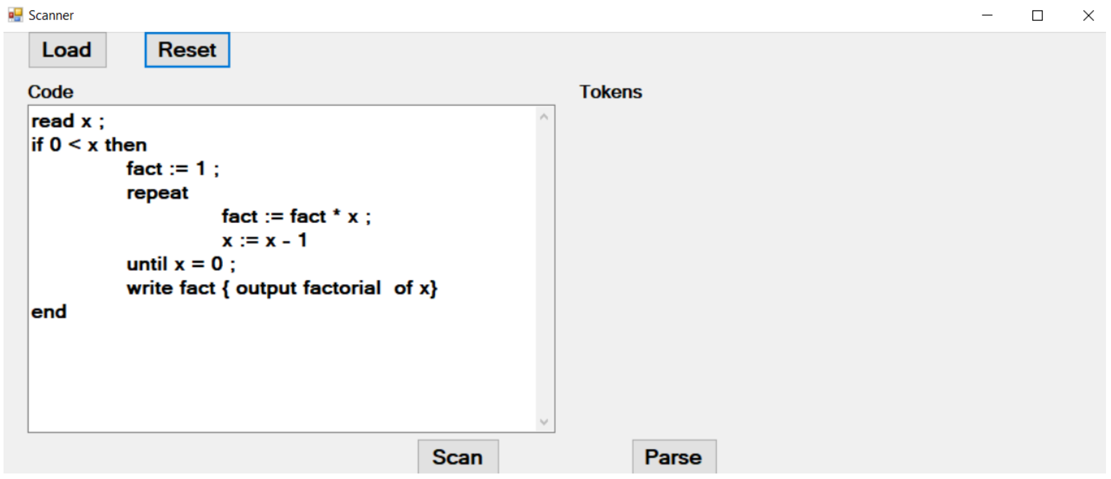
Then we can add any text file we need by pressing on “Load” button.



Name	Date modified	Type
Scanner	12/14/2021 12:54 AM	File f
Code.txt	12/16/2021 8:41 AM	Text I

Figure 2 Load button

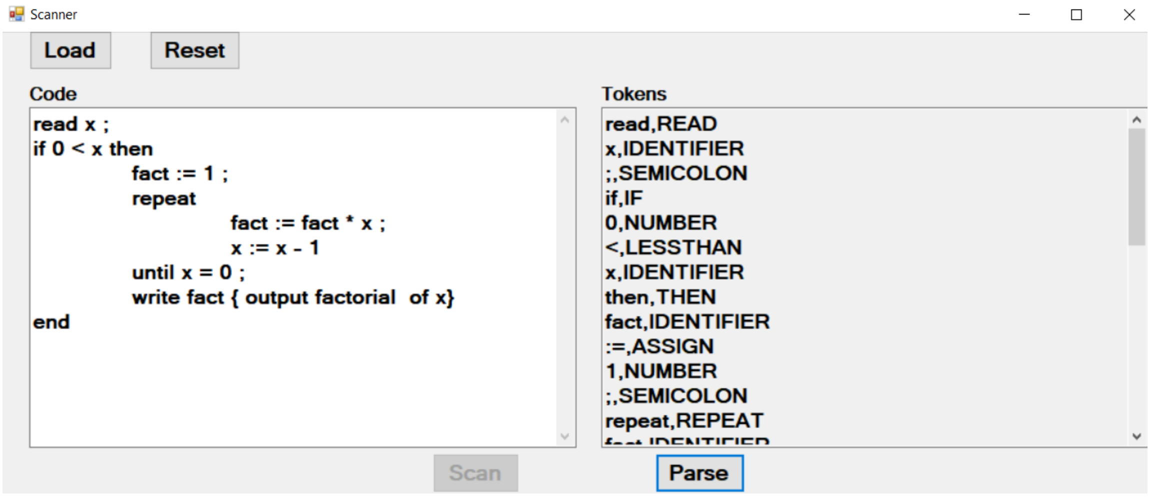
We will see that the context of file we loaded will appear in code textbox.



```
read x ;
if 0 < x then
          fact := 1 ;
          repeat
                    fact := fact * x ;
                    x := x - 1
          until x = 0 ;
          write fact { output factorial  of x}
end
```

Figure 3 GUI after add text file

if we press scan to define each part of code

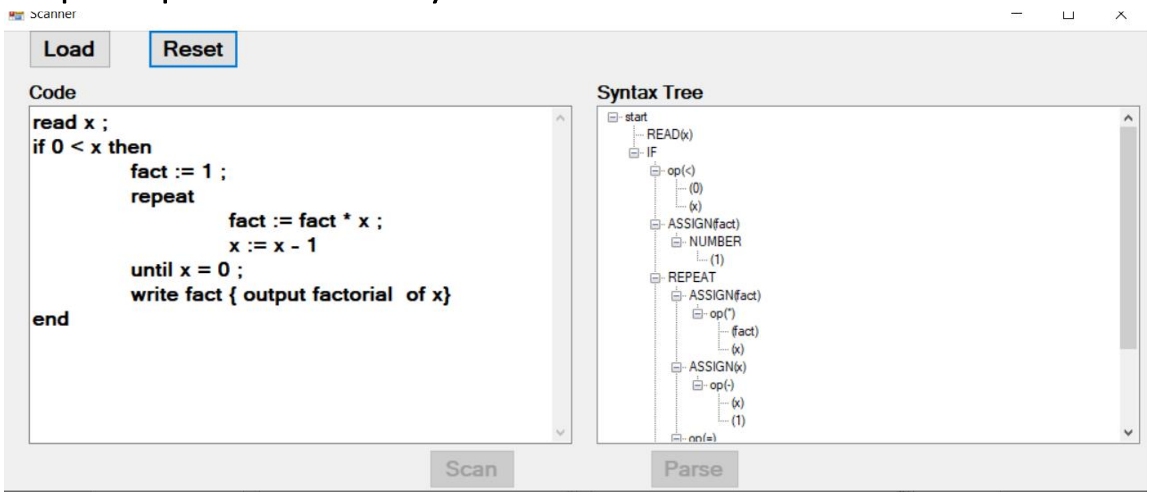


```
read x ;
if 0 < x then
          fact := 1 ;
          repeat
                    fact := fact * x ;
                    x := x - 1
          until x = 0 ;
          write fact { output factorial  of x}
end
```

```
read,READ
x,IDENTIFIER
;,SEMICOLON
if,IF
0,NUMBER
<,LESSTHAN
x,IDENTIFIER
then,THEN
fact,IDENTIFIER
:=,ASSIGN
1,NUMBER
;,SEMICOLON
repeat,REPEAT
```

Figure 4 GUI after press scan button

if we press parse to draw syntax tree.

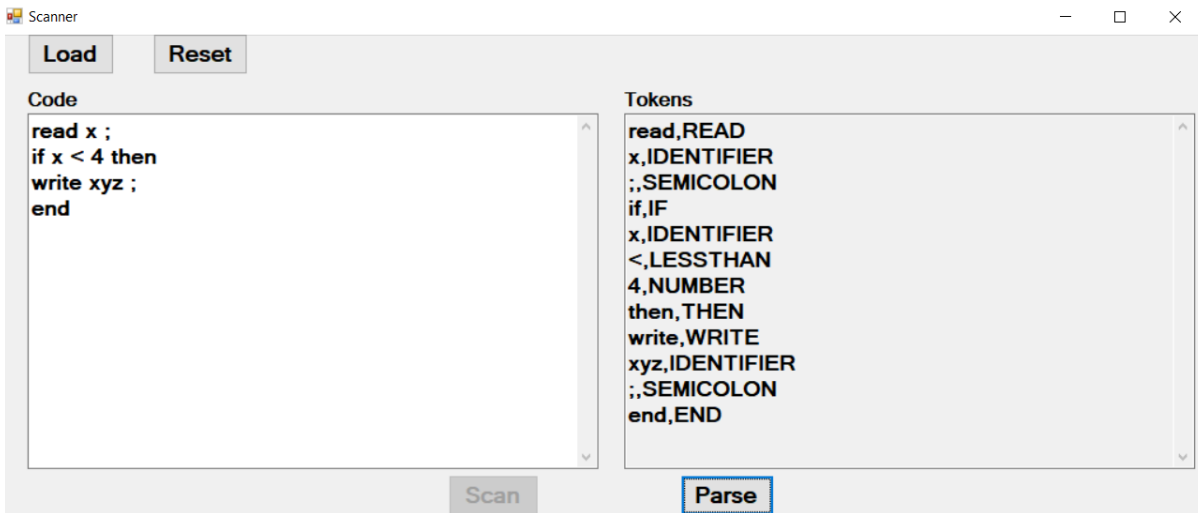


```
read x ;
if 0 < x then
          fact := 1 ;
          repeat
                    fact := fact * x ;
                    x := x - 1
          until x = 0 ;
          write fact { output factorial  of x}
end
```

```
start
  READ(x)
  IF
    op(<)
      (0)
      (x)
    ASSIGN(fact)
      NUMBER
        (1)
    REPEAT
      ASSIGN(fact)
        op(*)
          (fact)
          (x)
      ASSIGN(x)
        op(-)
          (x)
          (1)
      op(=)
```

For Ex2

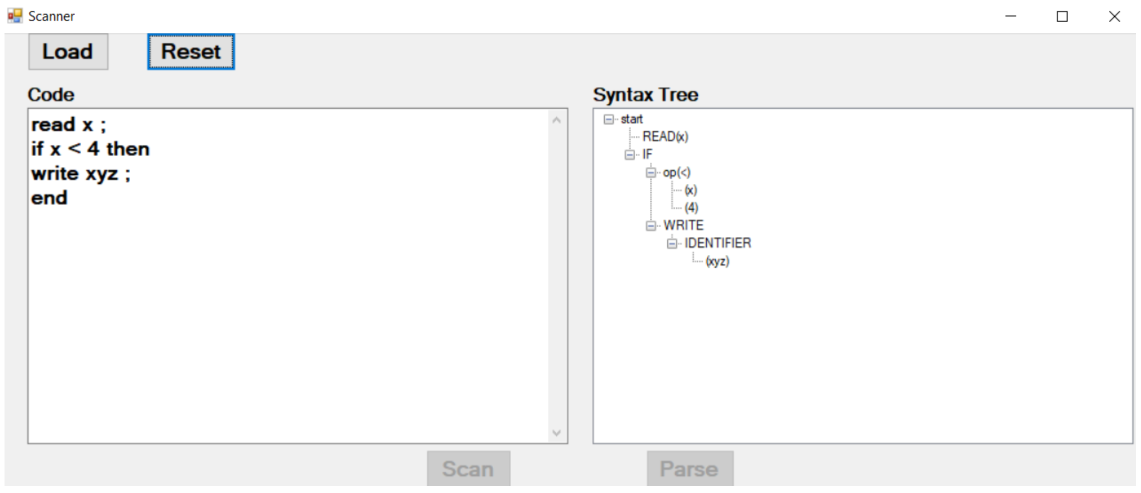
Scan



```
read x ;
if x < 4 then
write xyz ;
end
```

```
read,READ
x,IDENTIFIER
;,SEMICOLON
if,IF
x,IDENTIFIER
<,LESSTHAN
4,NUMBER
then,THEN
write,WRITE
xyz,IDENTIFIER
;,SEMICOLON
end,END
```

Parse

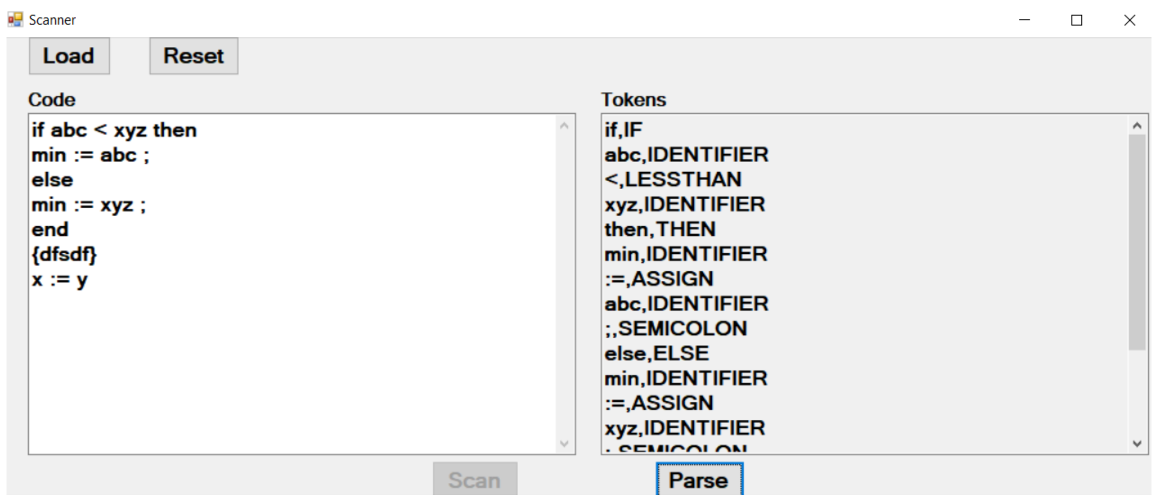


```
read x ;
if x < 4 then
write xyz ;
end
```

```
start
  READ(x)
  IF
    op(<)
      (x)
      (4)
    WRITE
      IDENTIFIER
        (xyz)
```

For Ex3

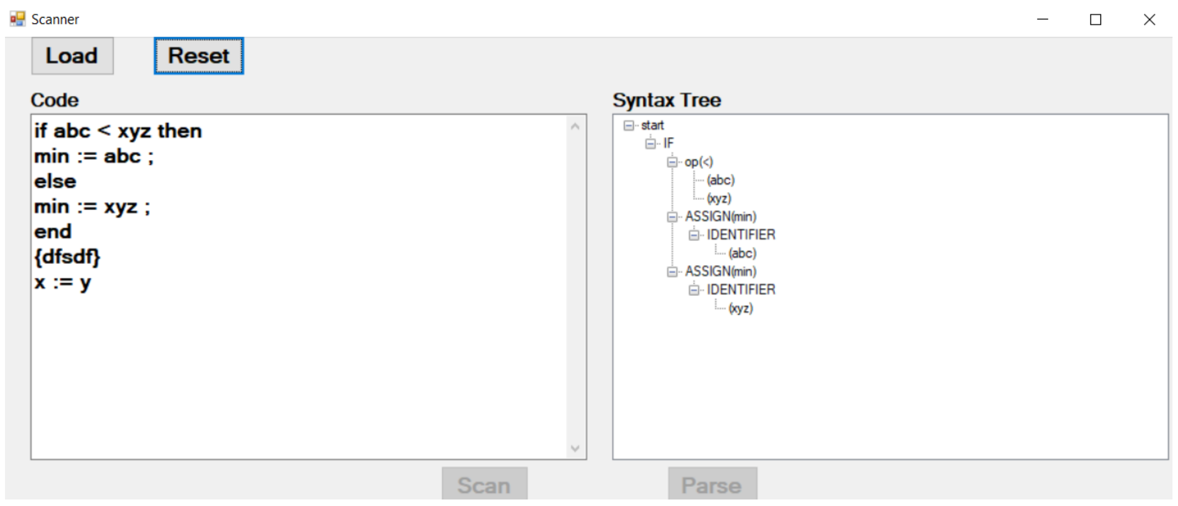
Scan



```
if abc < xyz then
min := abc ;
else
min := xyz ;
end
{dfsdf}
x := y
```

```
if,IF
abc,IDENTIFIER
<,LESSTHAN
xyz,IDENTIFIER
then,THEN
min,IDENTIFIER
:=,ASSIGN
abc,IDENTIFIER
;,SEMICOLON
else,ELSE
min,IDENTIFIER
:=,ASSIGN
xyz,IDENTIFIER
;,SEMICOLON
```

Parse

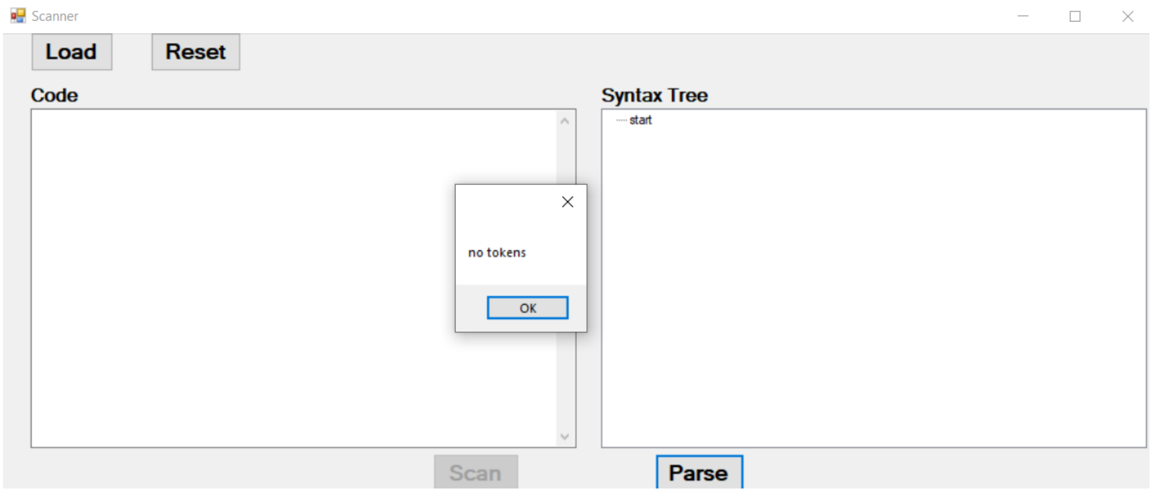


```
if abc < xyz then
min := abc ;
else
min := xyz ;
end
{dfsdf}
x := y
```

```
start
  IF
    op(<)
      (abc)
      (xyz)
    ASSIGN(min)
      IDENTIFIER
        (abc)
    ASSIGN(min)
      IDENTIFIER
        (xyz)
```

Check Errors

1- For no code and press parse
So the program will display a message “No tokens”

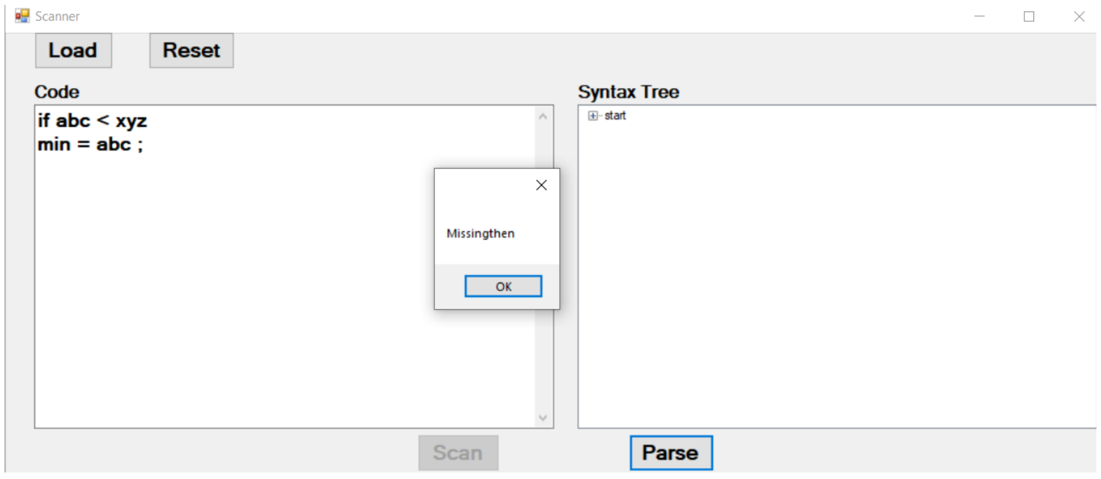


```
no tokens
```

2- Check the syntax error of the tiny language
So, with using this code

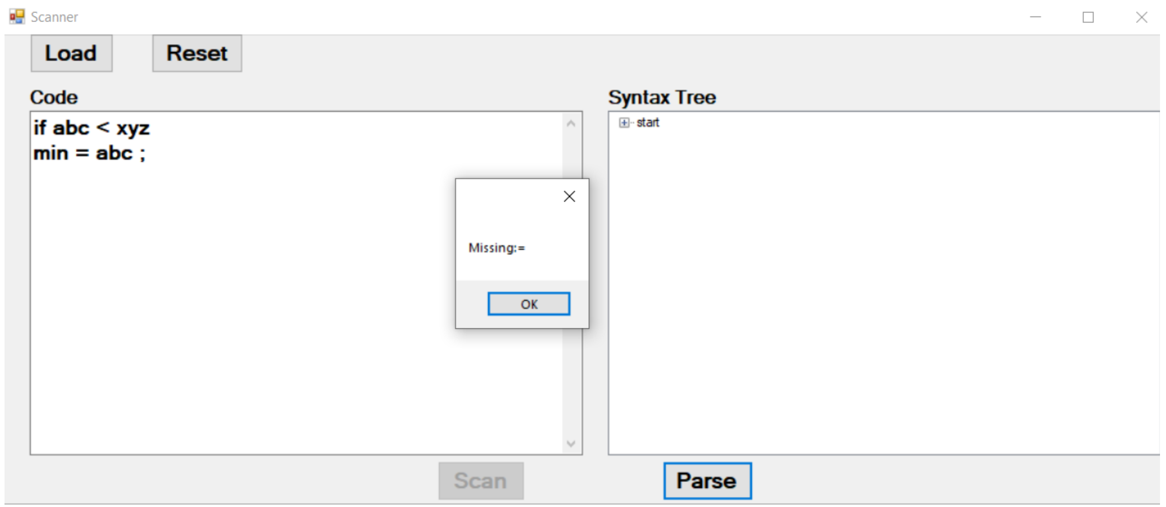
```
if abc < xyz
min = abc ;
```

- here in this code as we studied that before each = we need to put “:”.
- for each if condition we need to put “then” and “end” at the end of the code of if.



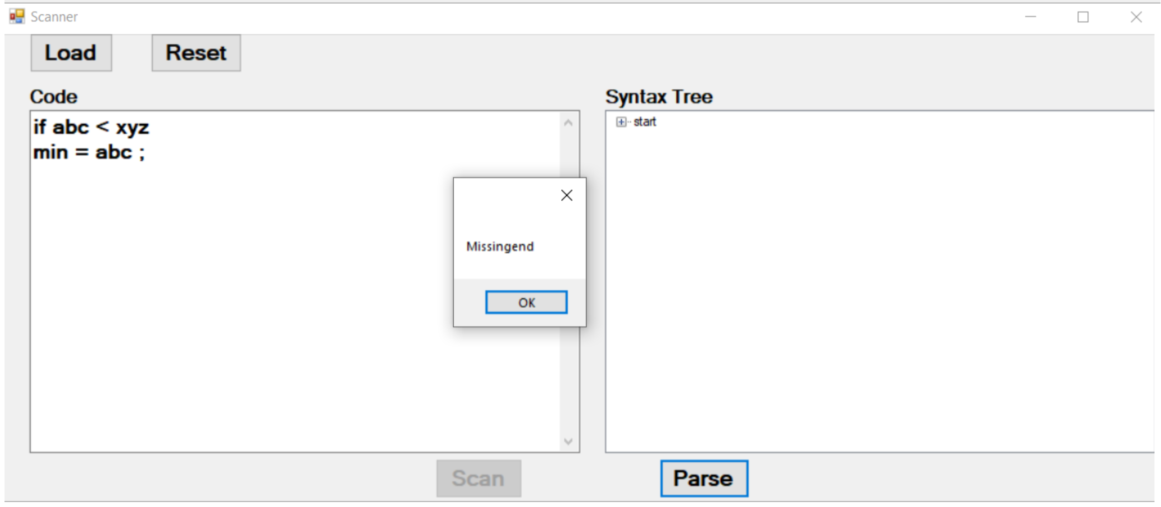
```
if abc < xyz
min = abc ;
```

```
Missingthen
```



```
if abc < xyz
min = abc ;
```

Missing:=



```
if abc < xyz
min = abc ;
```

Missingend

Drive Link