AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING

Project Compilers

ID	*Name*
2000318	*Karim Mikhail Louis Elnekheily*
2000923	*Mark Ramy Fathy*
2001400	*Heba Maher Abdelrahman*
2001440	*Youssef Saad Gobran*
2000003	*Andrew Ayman Samir*
2000955	*Philopateer Sameh Rasmy*

Project Release containing EXE :

Release project

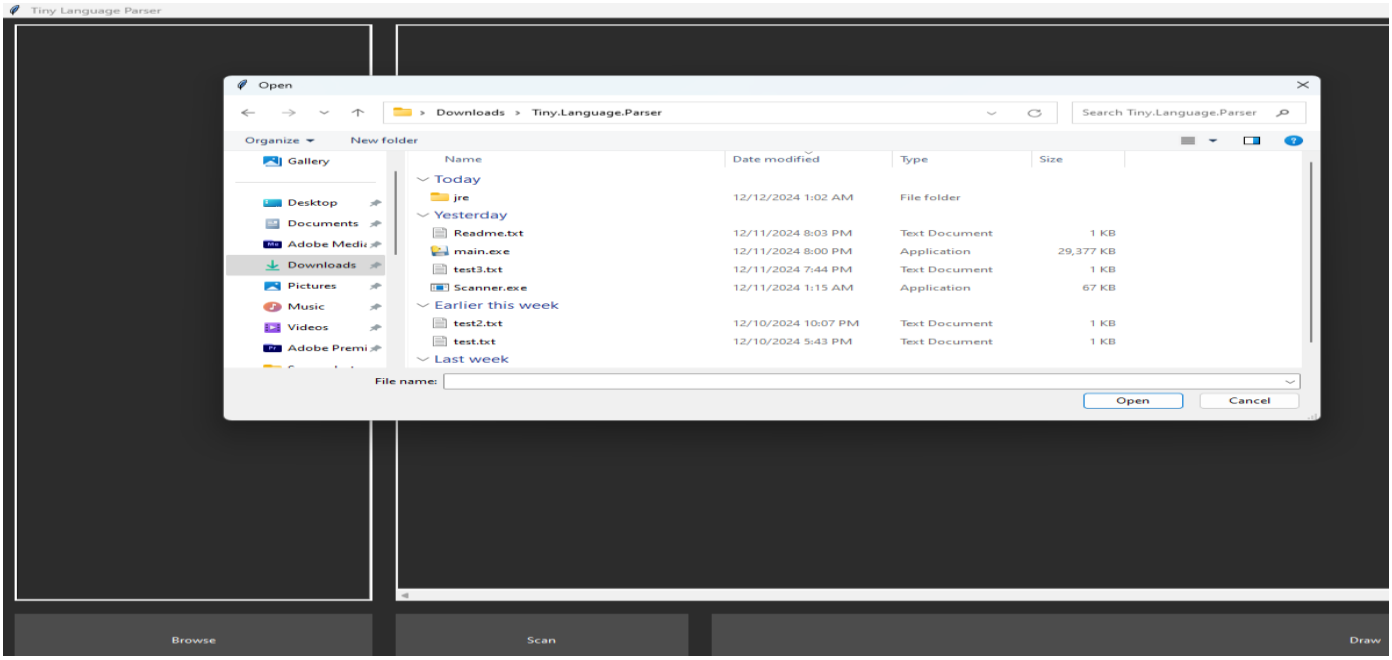
Github Repository:

Project Repository

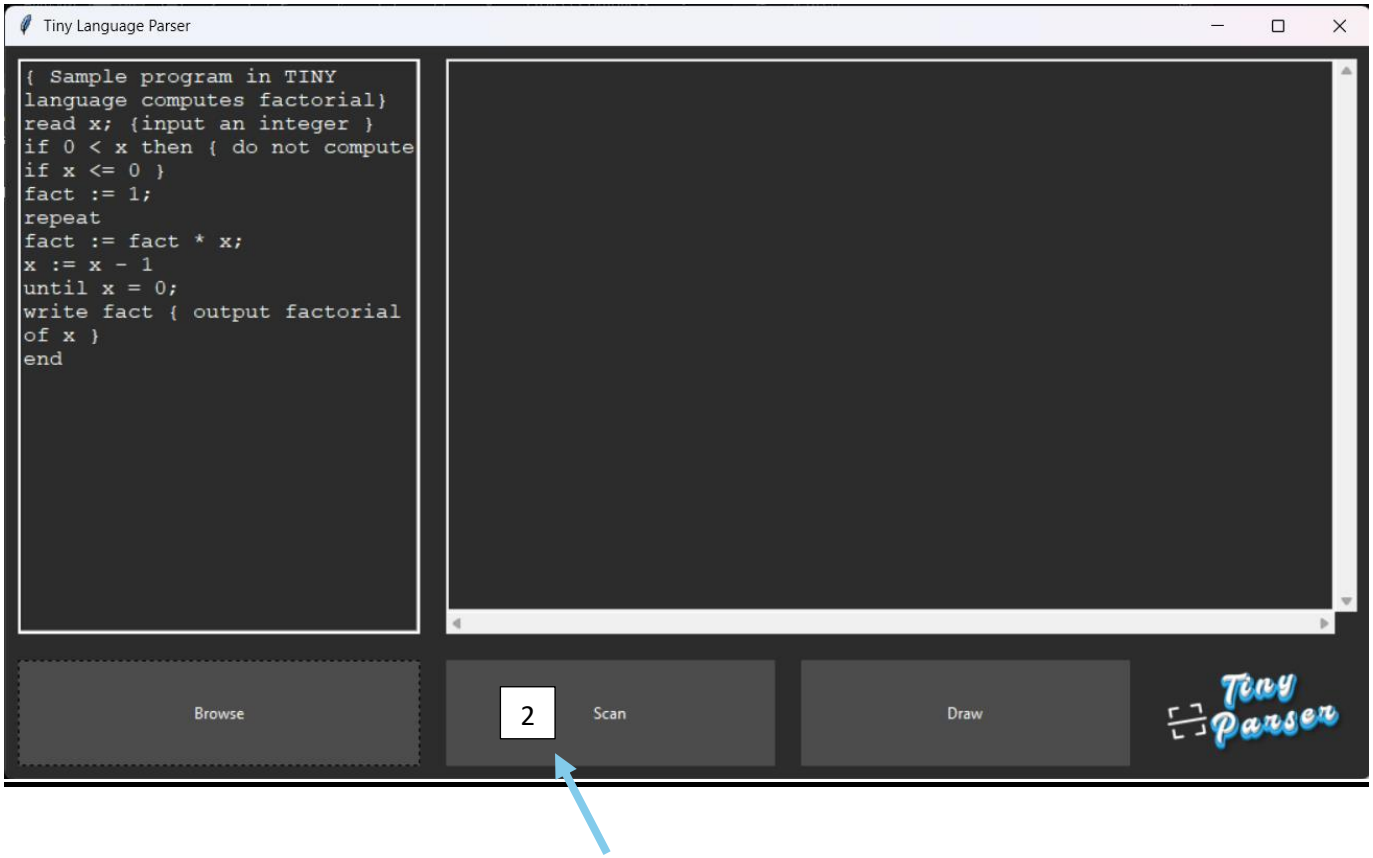
Screenshots Examples :

Example1:

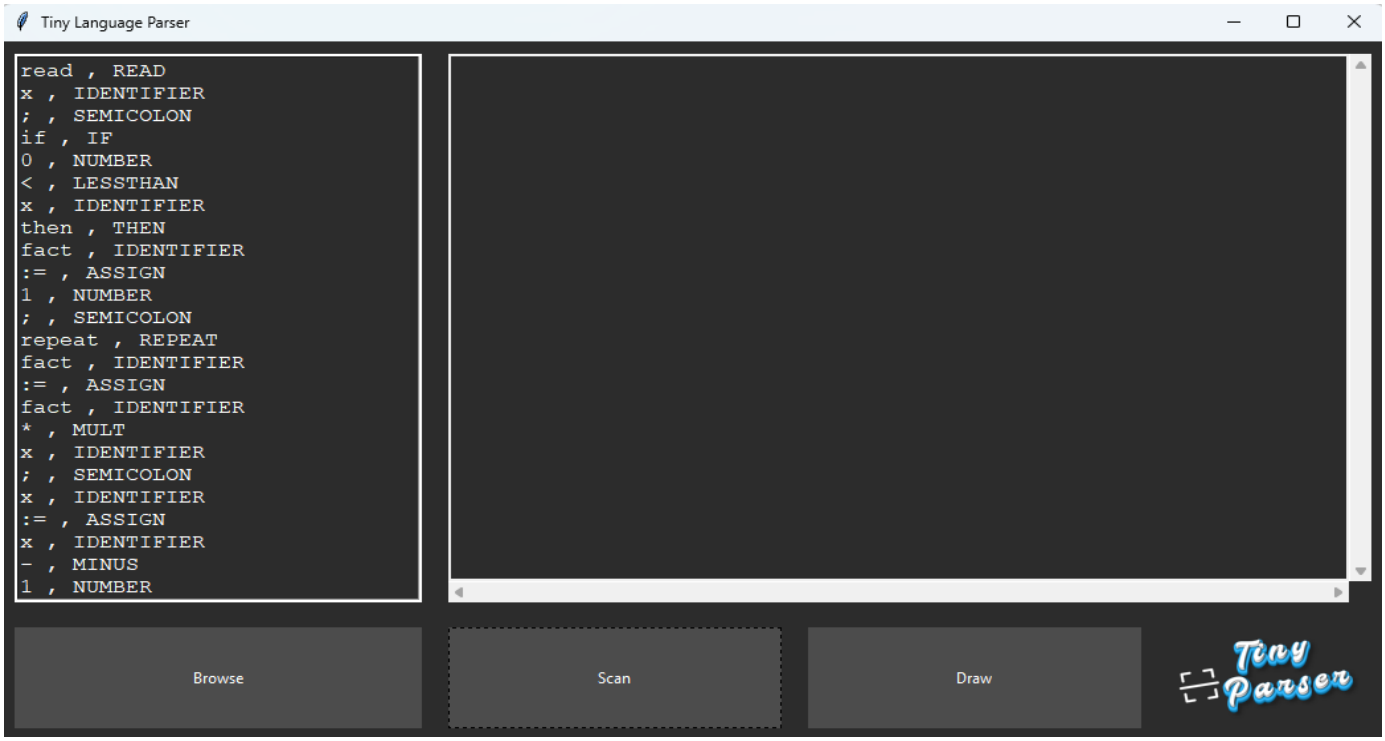
First we browse the file:



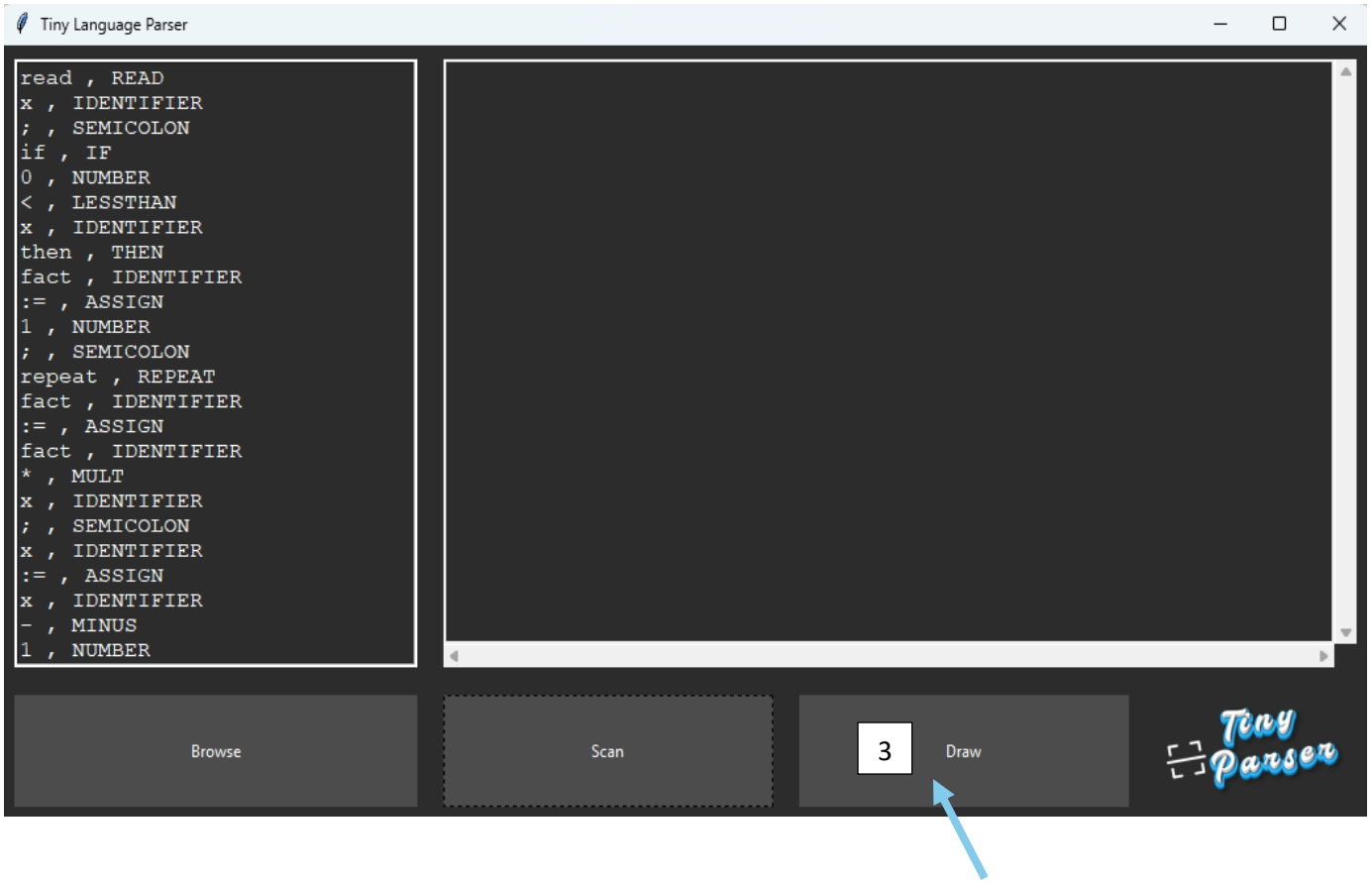
Then we scan the file

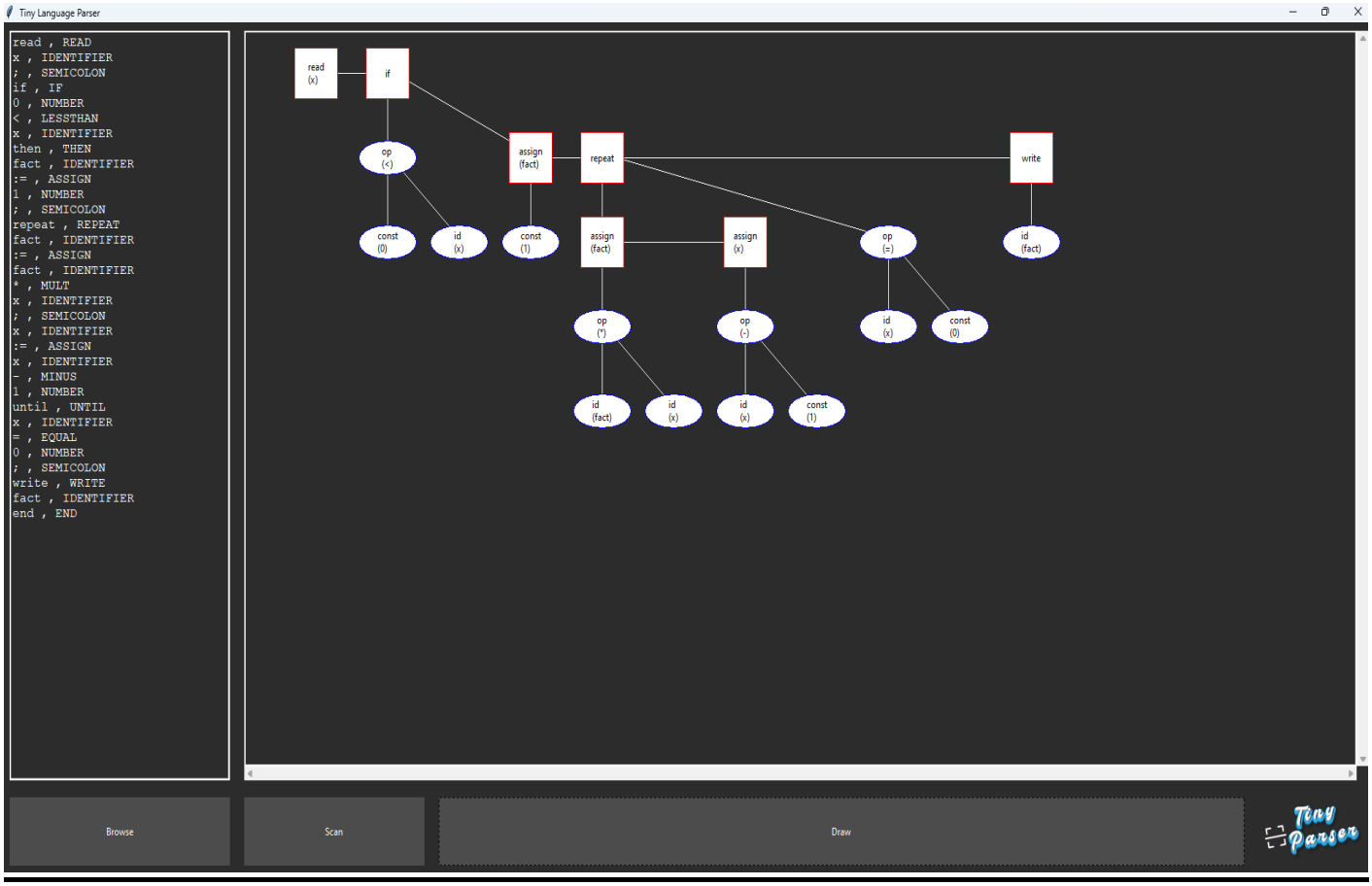


After Scanning:

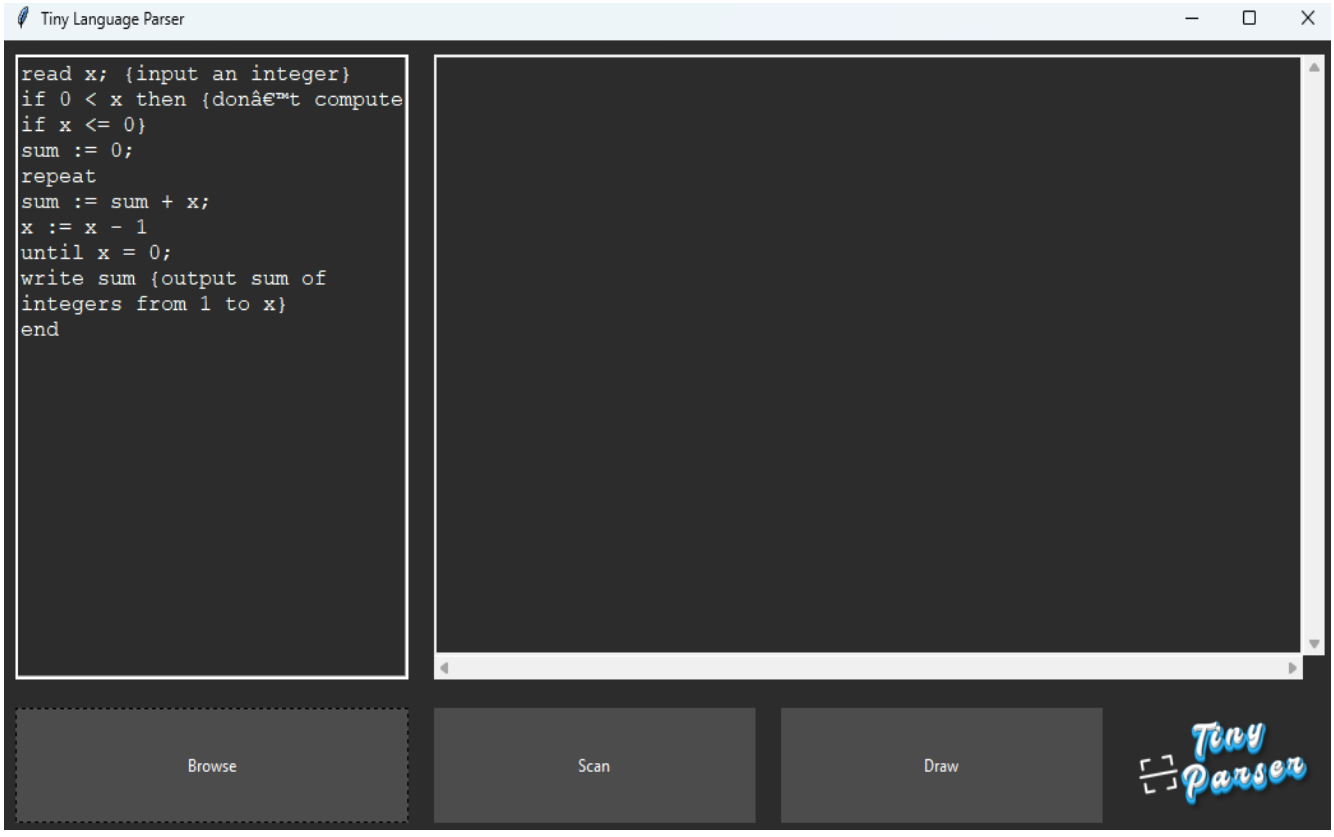

Tiny Language Parser
read , READ
x , IDENTIFIER
; , SEMICOLON
if , IF
0 , NUMBER
< , LESSTHAN
x , IDENTIFIER
then , THEN
fact , IDENTIFIER
:= , ASSIGN
1 , NUMBER
; , SEMICOLON
repeat , REPEAT
fact , IDENTIFIER
:= , ASSIGN
fact , IDENTIFIER
* , MULT
x , IDENTIFIER
; , SEMICOLON
x , IDENTIFIER
:= , ASSIGN
x , IDENTIFIER
- , MINUS
1 , NUMBER
Browse
Scan
Draw
Tiny Parser

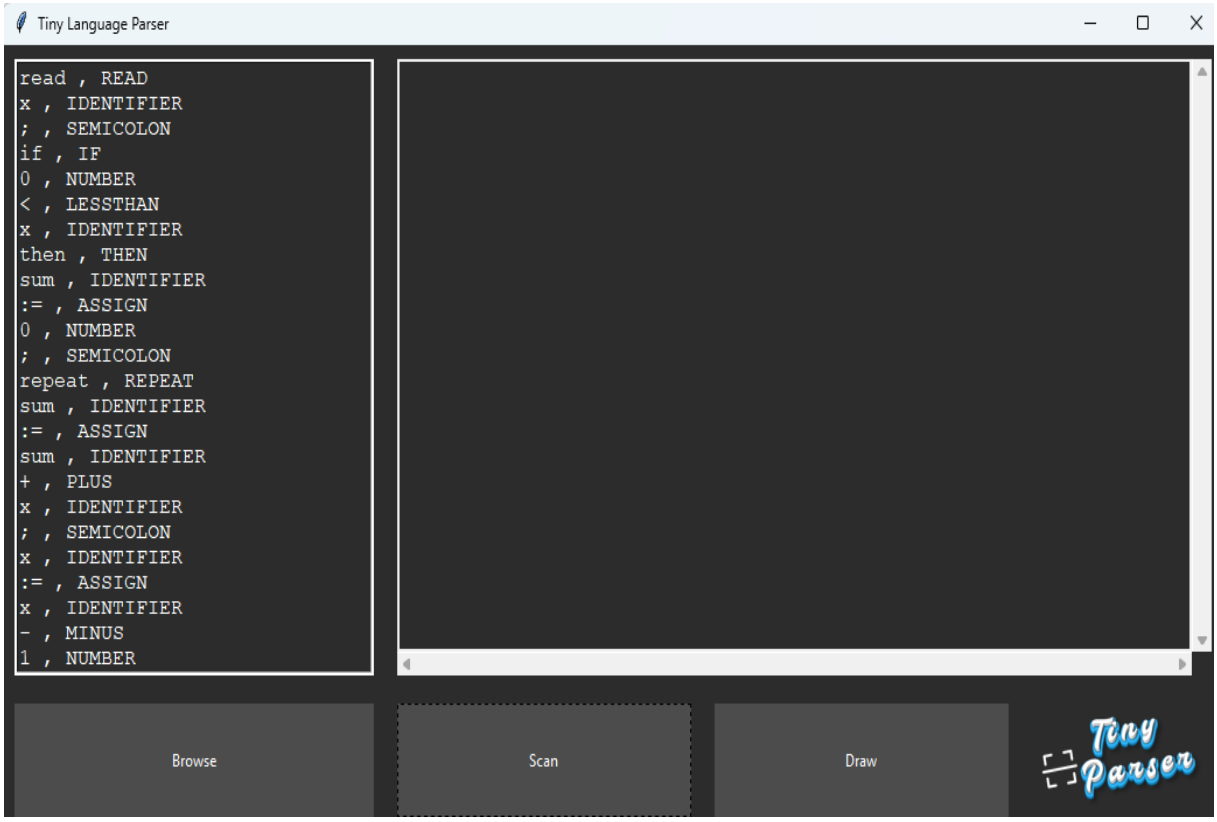
Parser:

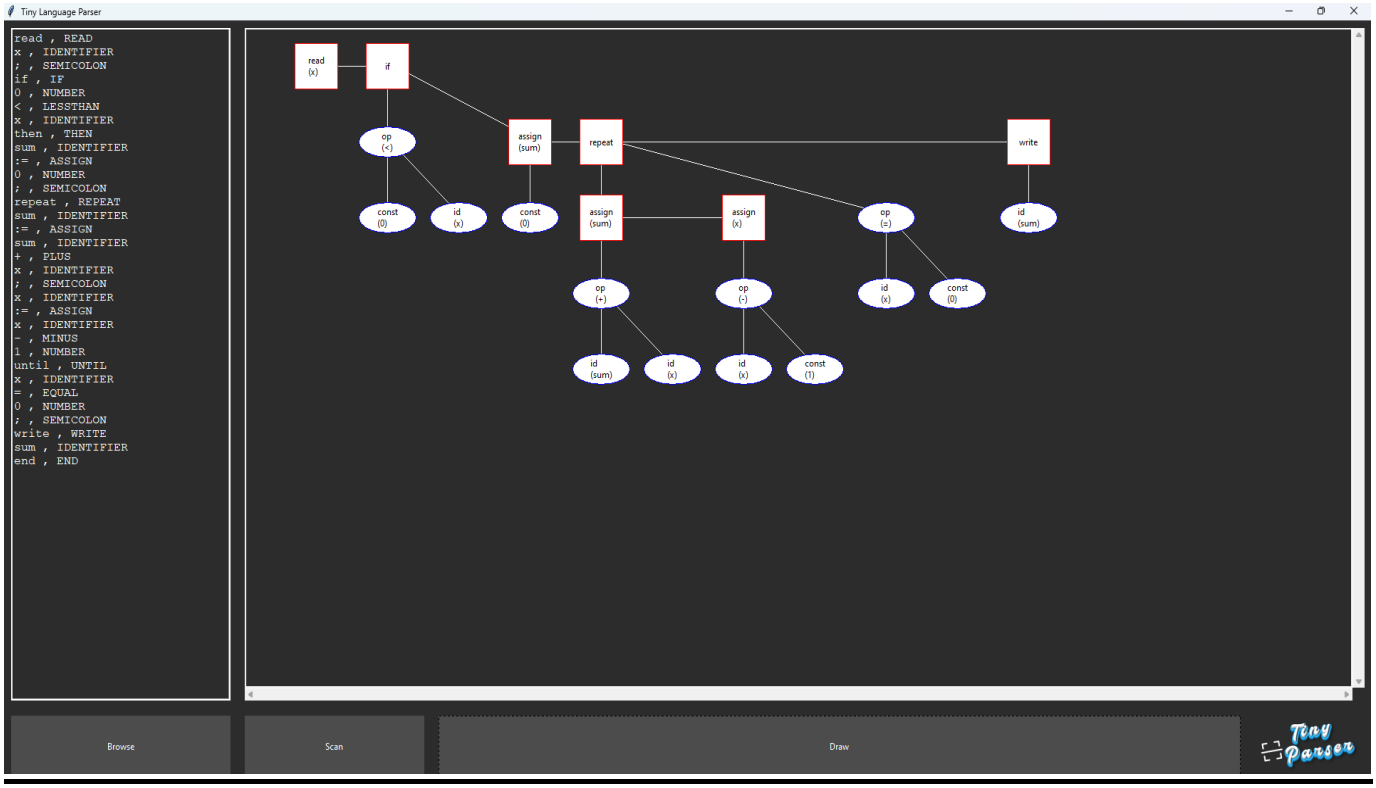

Tiny Language Parser
read , READ
x , IDENTIFIER
; , SEMICOLON
if , IF
0 , NUMBER
< , LESSTHAN
x , IDENTIFIER
then , THEN
fact , IDENTIFIER
:= , ASSIGN
1 , NUMBER
; , SEMICOLON
repeat , REPEAT
fact , IDENTIFIER
:= , ASSIGN
fact , IDENTIFIER
* , MULT
x , IDENTIFIER
; , SEMICOLON
x , IDENTIFIER
:= , ASSIGN
x , IDENTIFIER
- , MINUS
1 , NUMBER
Browse
Scan
3
Draw
Tiny Parser


Tiny Language Parser
read , READ
x , IDENTIFIER
; , SEMICOLON
if , IF
0 , NUMBER
< , LESSTHAN
x , IDENTIFIER
then , THEN
fact , IDENTIFIER
:= , ASSIGN
1 , NUMBER
; , SEMICOLON
repeat , REPEAT
fact , IDENTIFIER
:= , ASSIGN
fact , IDENTIFIER
* , MULT
x , IDENTIFIER
; , SEMICOLON
x , IDENTIFIER
:= , ASSIGN
x , IDENTIFIER
- , MINUS
1 , NUMBER
until , UNTIL
x , IDENTIFIER
= , EQUAL
0 , NUMBER
; , SEMICOLON
write , WRITE
fact , IDENTIFIER
end , END
read (x)
if
op (<)
assign (fact)
repeat
write
const (0)
id (x)
const (1)
assign (fact)
assign (x)
op (=)
id (fact)
op (*)
op (-)
id (x)
const (0)
id (fact)
id (x)
id (x)
const (1)
Browse
Scan
Draw
Tiny Parser

Example 2:

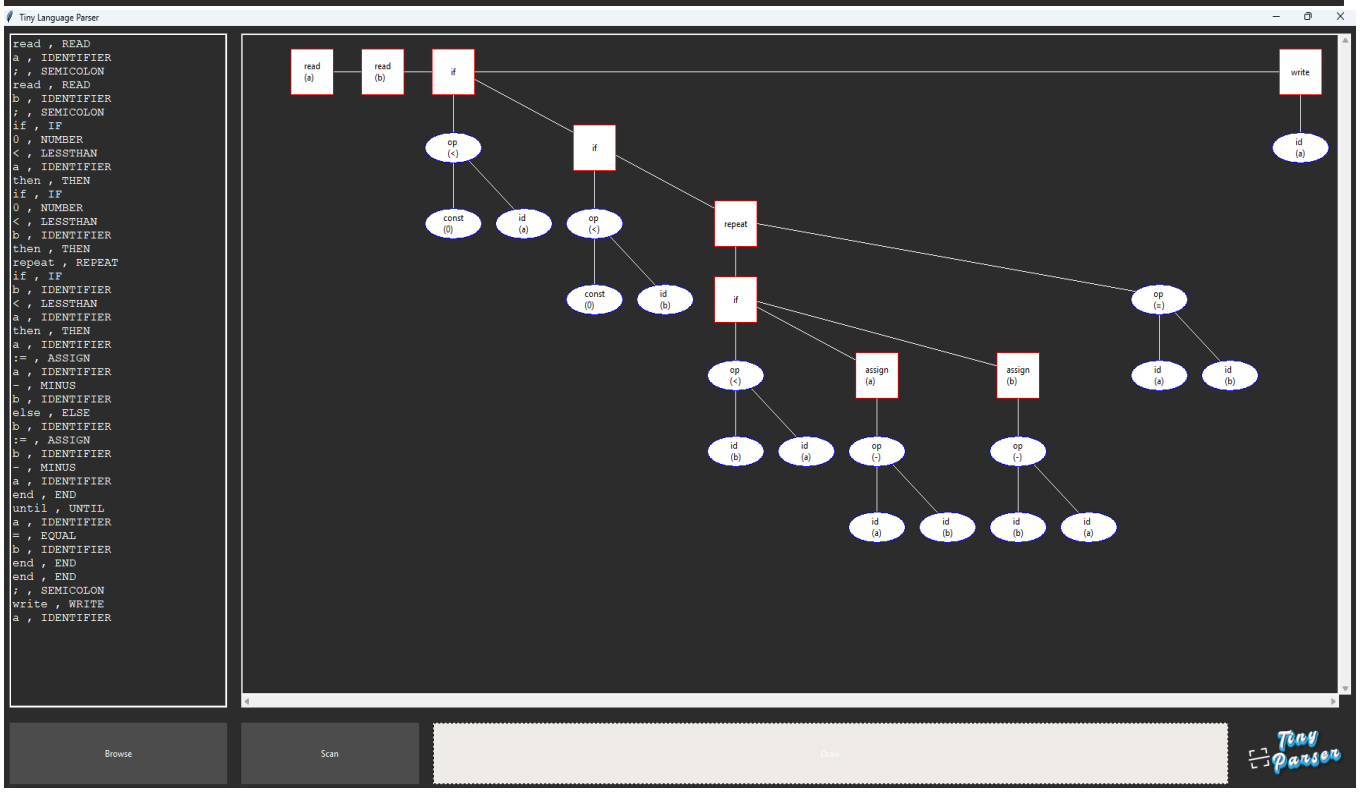

Tiny Language Parser
read x; {input an integer}
if 0 < x then {donâ€™t compute
if x <= 0}
sum := 0;
repeat
sum := sum + x;
x := x - 1
until x = 0;
write sum {output sum of
integers from 1 to x}
end
Browse
Scan
Draw
Tiny Parser


Tiny Language Parser
read , READ
x , IDENTIFIER
; , SEMICOLON
if , IF
0 , NUMBER
< , LESSTHAN
x , IDENTIFIER
then , THEN
sum , IDENTIFIER
:= , ASSIGN
0 , NUMBER
; , SEMICOLON
repeat , REPEAT
sum , IDENTIFIER
:= , ASSIGN
sum , IDENTIFIER
+ , PLUS
x , IDENTIFIER
; , SEMICOLON
x , IDENTIFIER
:= , ASSIGN
x , IDENTIFIER
- , MINUS
1 , NUMBER
Browse
Scan
Draw
Tiny Parser


Tiny Language Parser
read , READ
x , IDENTIFIER
; , SEMICOLON
if , IF
0 , NUMBER
< , LESSTHAN
x , IDENTIFIER
then , THEN
sum , IDENTIFIER
:= , ASSIGN
0 , NUMBER
; , SEMICOLON
repeat , REPEAT
sum , IDENTIFIER
:= , ASSIGN
sum , IDENTIFIER
+ , PLUS
x , IDENTIFIER
; , SEMICOLON
x , IDENTIFIER
:= , ASSIGN
x , IDENTIFIER
- , MINUS
1 , NUMBER
until , UNTIL
x , IDENTIFIER
= , EQUAL
0 , NUMBER
; , SEMICOLON
write , WRITE
sum , IDENTIFIER
end , END
read (x)
if
op (<)
assign (sum)
repeat
write
const (0)
id (x)
const (0)
assign (sum)
assign (x)
op (=)
id (sum)
op (+)
op (-)
id (x)
const (0)
id (sum)
id (x)
id (x)
const (1)
Browse
Scan
Draw
Tiny Parser

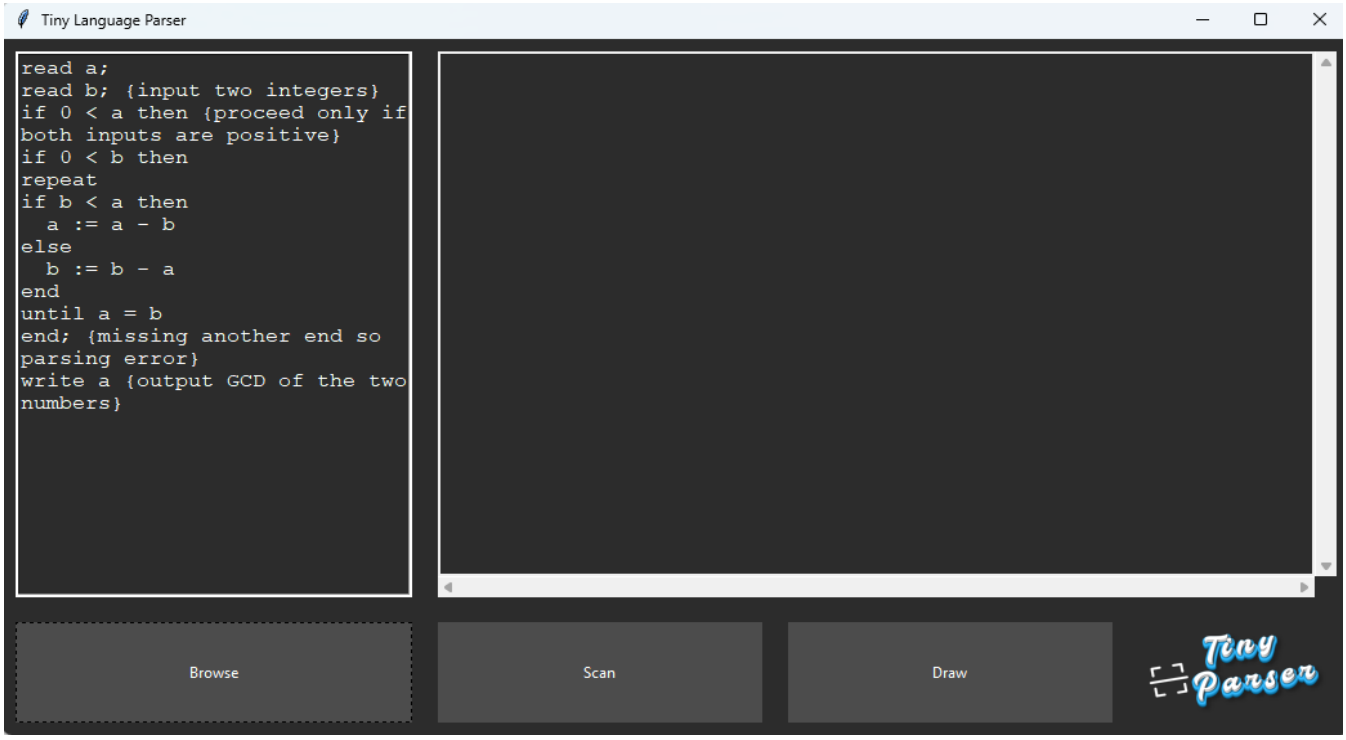
Example 3:

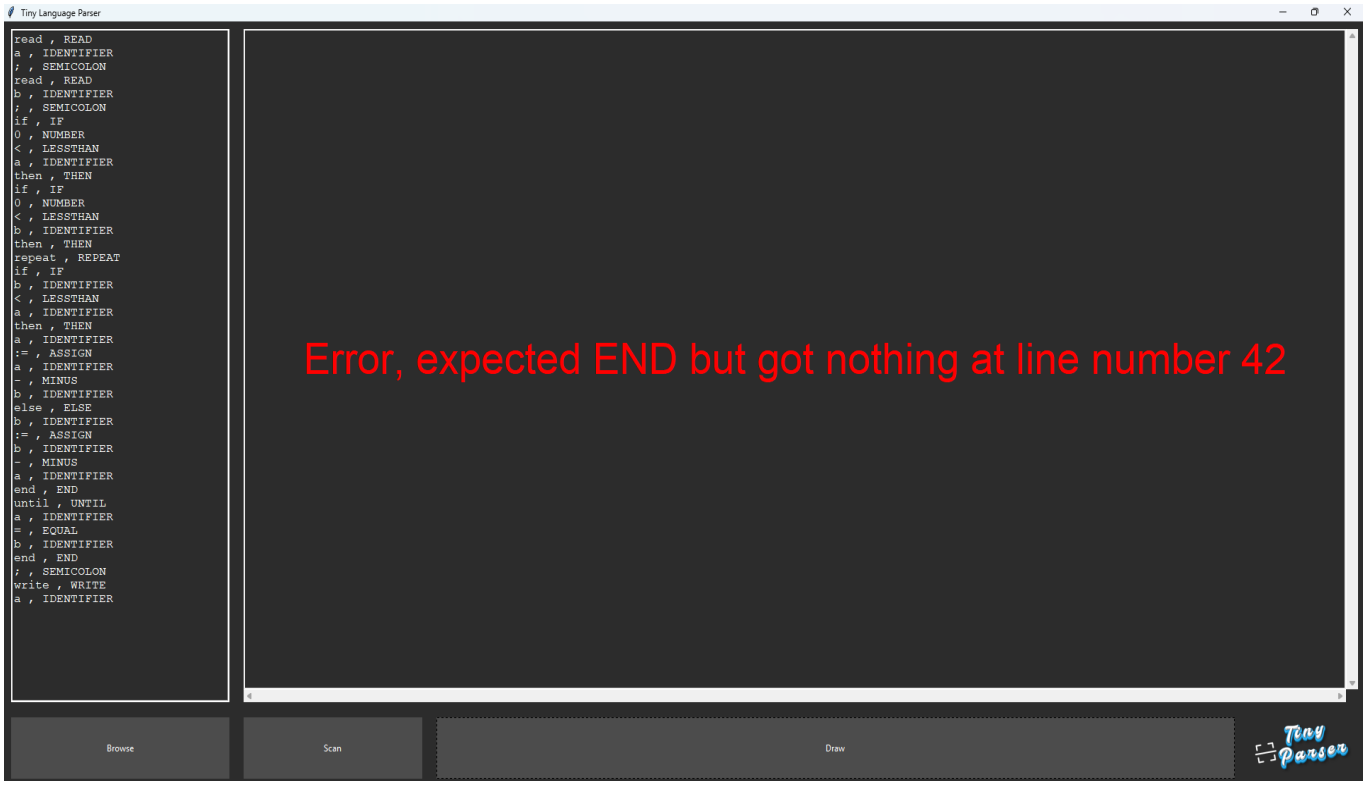
Tiny Language Parser
read a;
read b; {input two integers}
if 0 < a then {proceed only if
both inputs are positive}
if 0 < b then
repeat
if b < a then
a := a - b
else
b := b - a
end
until a = b
end
end;
write a {output GCD of the two
numbers}
Browse
Scan
Draw
Tiny Parser


Tiny Language Parser
read , READ
a , IDENTIFIER
; , SEMICOLON
read , READ
b , IDENTIFIER
; , SEMICOLON
if , IF
0 , NUMBER
< , LESSTHAN
a , IDENTIFIER
then , THEN
if , IF
0 , NUMBER
< , LESSTHAN
b , IDENTIFIER
then , THEN
repeat , REPEAT
if , IF
b , IDENTIFIER
< , LESSTHAN
a , IDENTIFIER
then , THEN
a , IDENTIFIER
:= , ASSIGN
a , IDENTIFIER
- , MINUS
b , IDENTIFIER
else , ELSE
b , IDENTIFIER
:= , ASSIGN
b , IDENTIFIER
- , MINUS
a , IDENTIFIER
end , END
until , UNTIL
a , IDENTIFIER
= , EQUAL
b , IDENTIFIER
end , END
end , END
; , SEMICOLON
write , WRITE
a , IDENTIFIER
read (a)
read (b)
if
write
id (a)
op (<)
if
const (0)
id (a)
op (<)
repeat
const (0)
id (b)
if
op (=)
op (<)
assign (a)
assign (b)
id (a)
id (b)
id (b)
id (a)
op (-)
op (-)
id (a)
id (b)
id (b)
id (a)
Browse
Scan
Draw
Tiny Parser

Example 4:

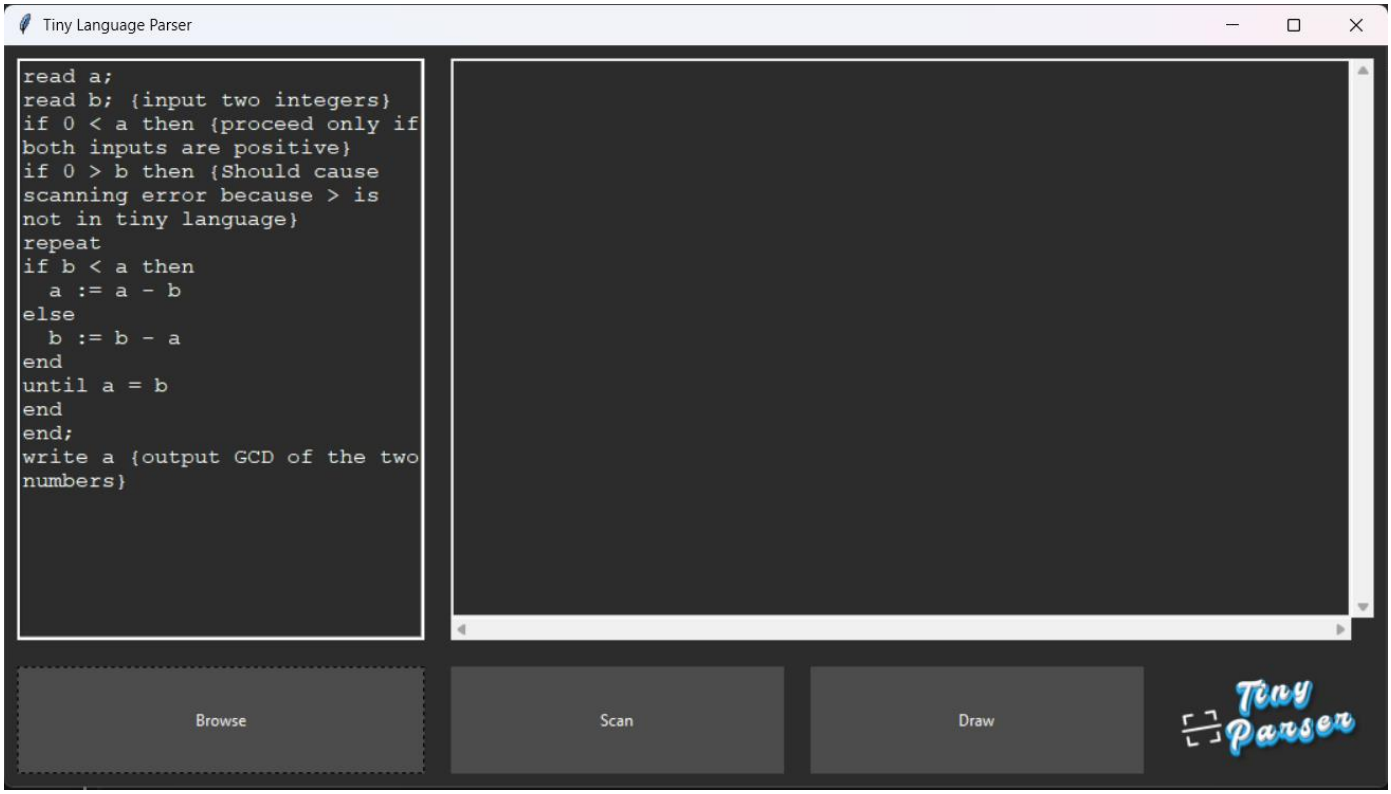
- Showing a parsing error due to a missing (end) in the code file

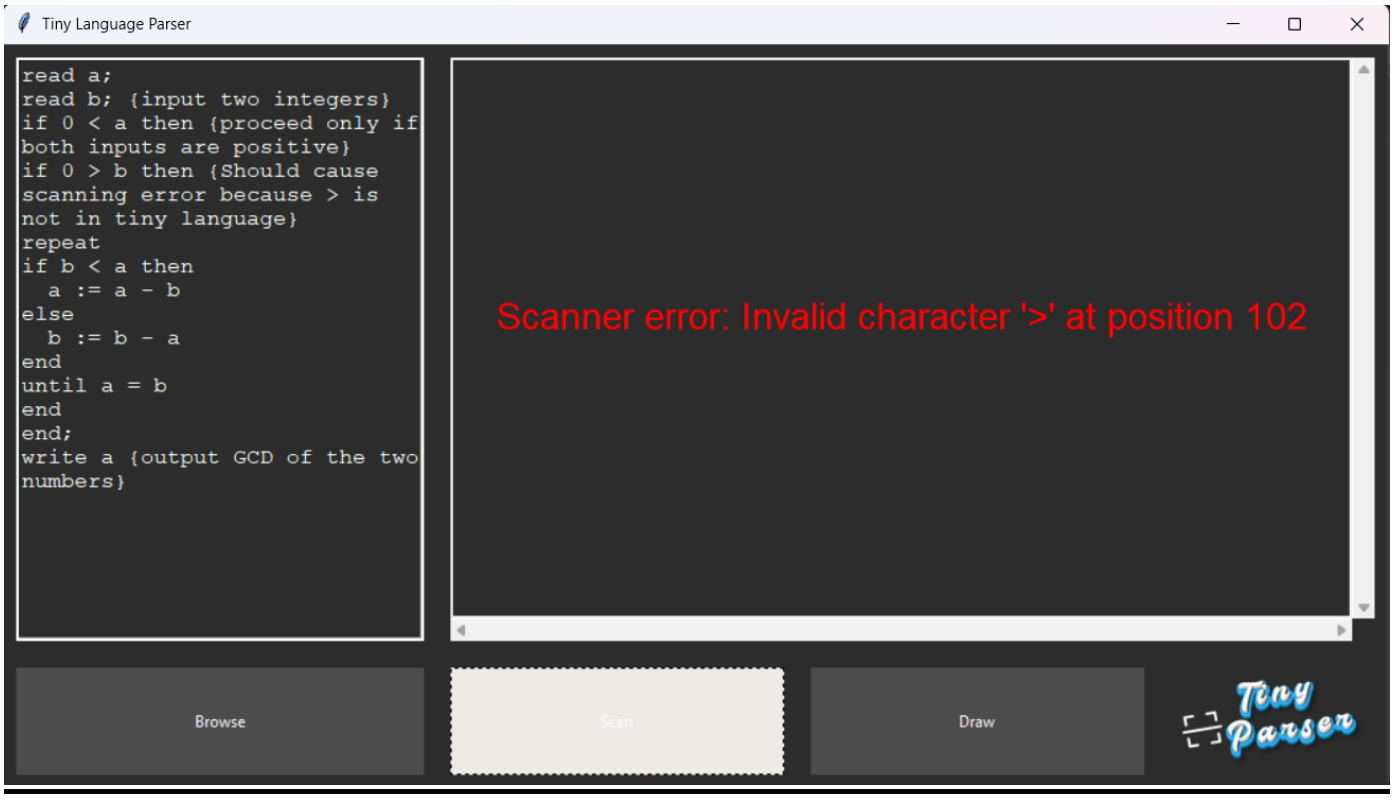




Example 5:

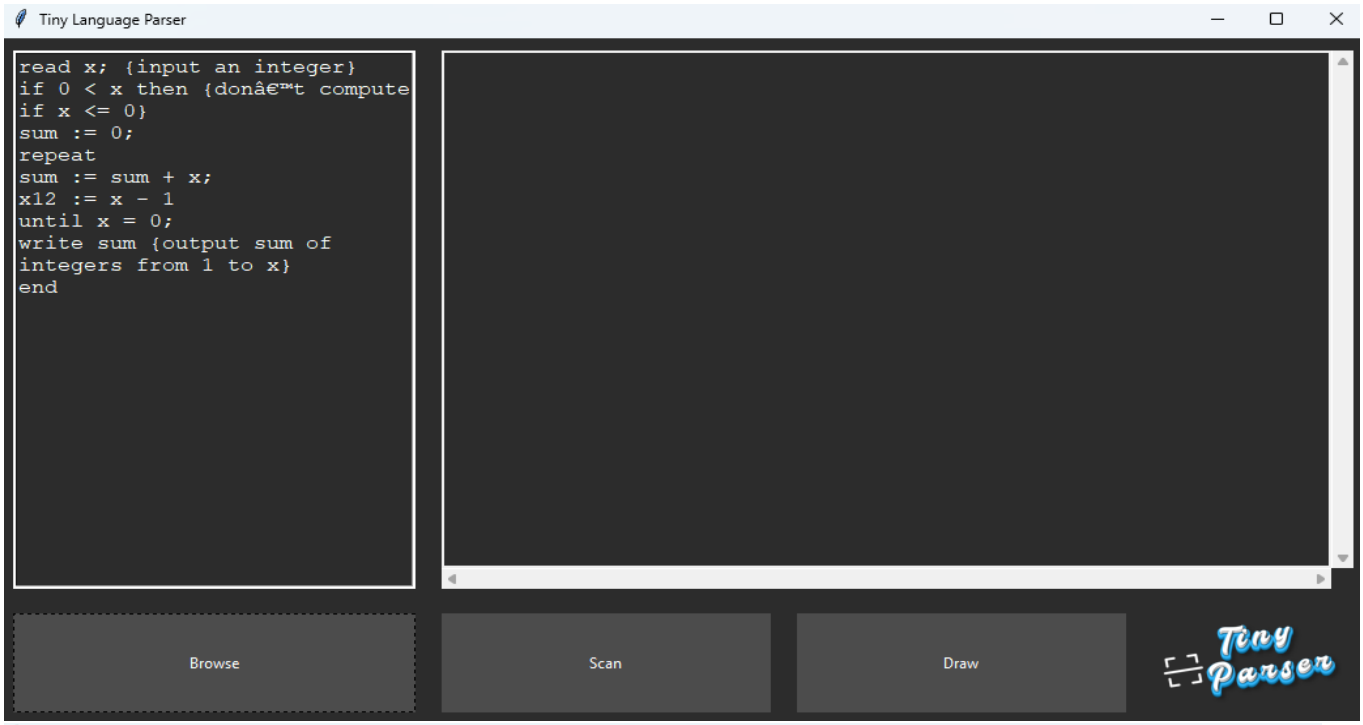
- Showing a scanning error due to an unrecognized character (>) in the code



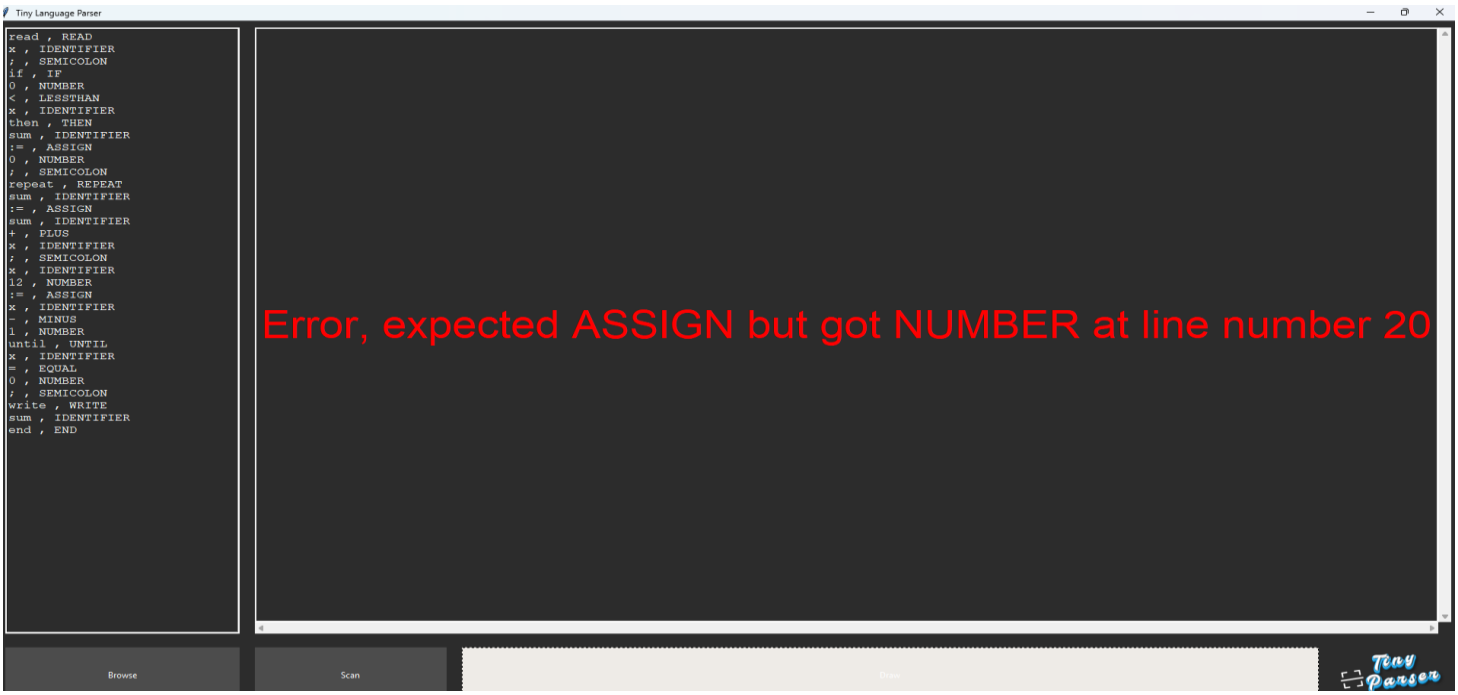


Example 6:

- Showing that it scans correctly X12 and it is not an error

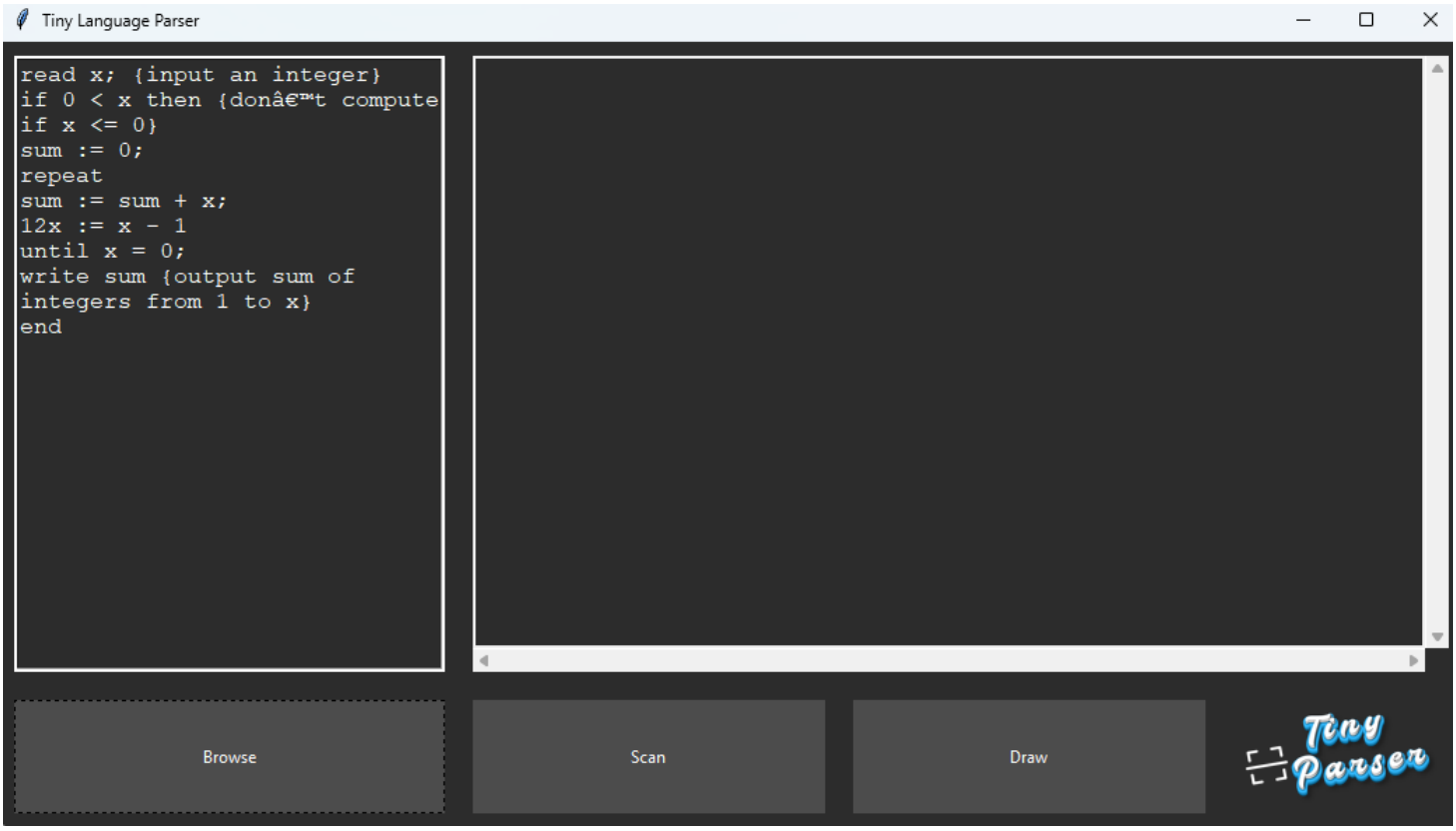


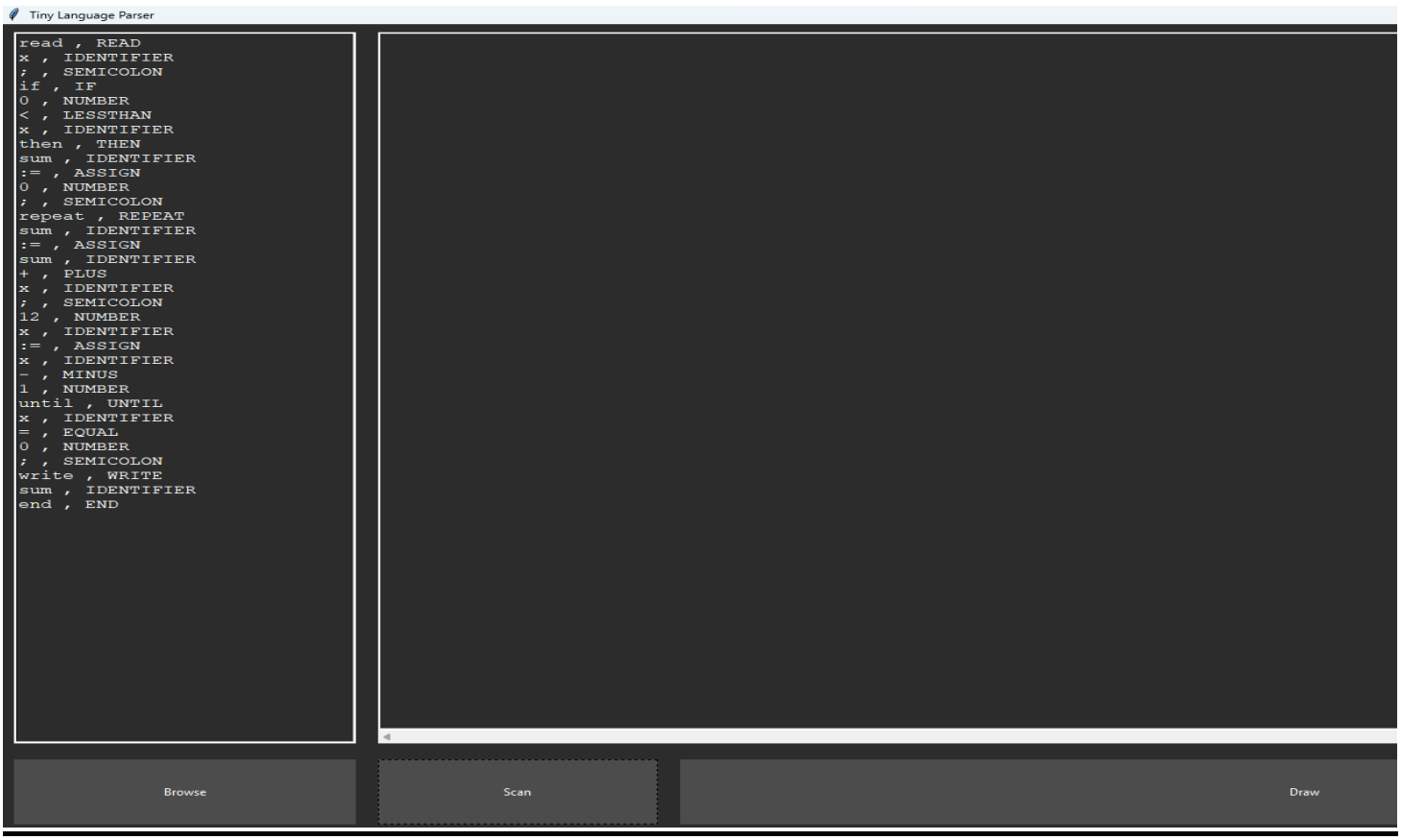
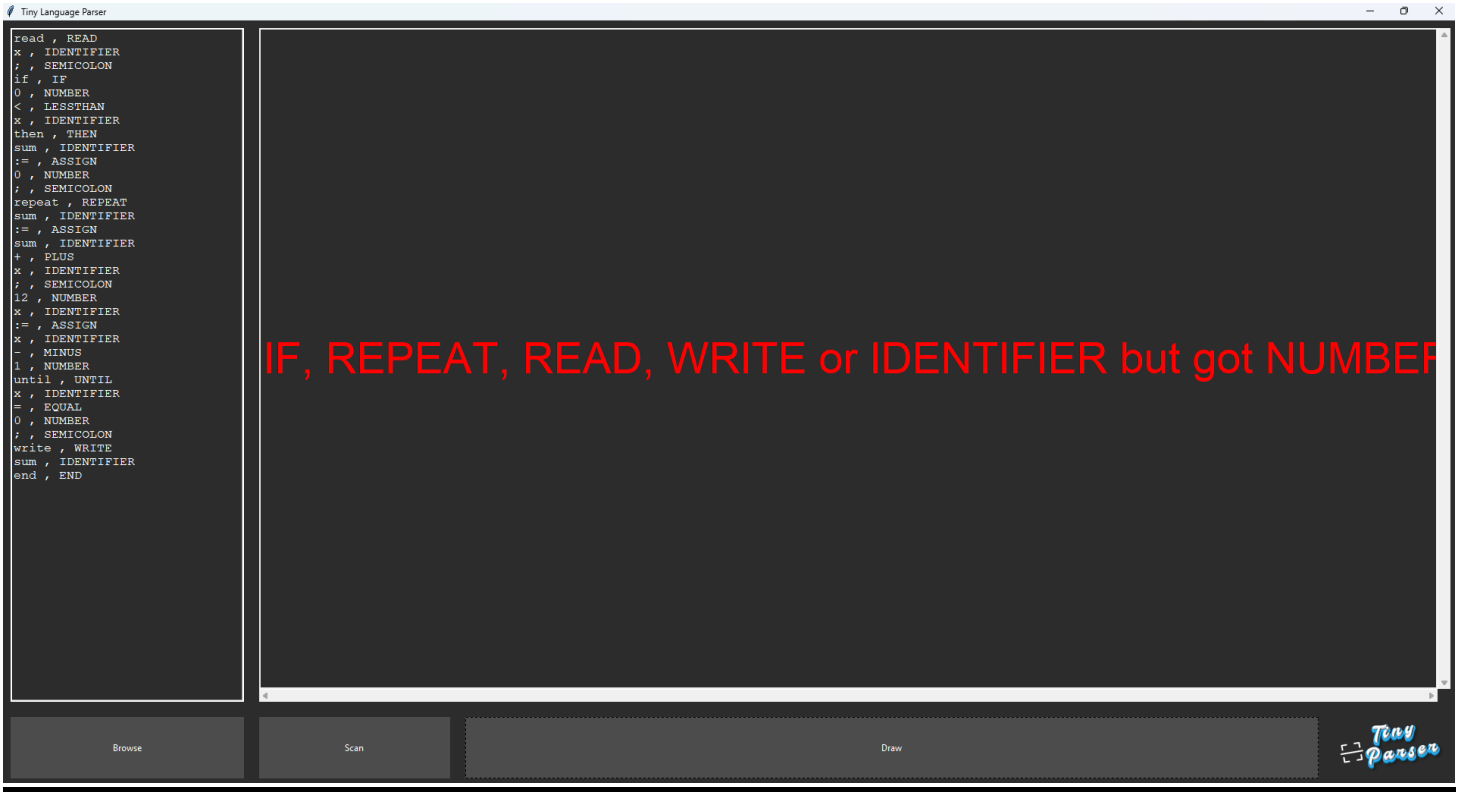
- Parser error:



Example 7:

- Showing that it scans correctly 12X and it is not an error





Parser Error