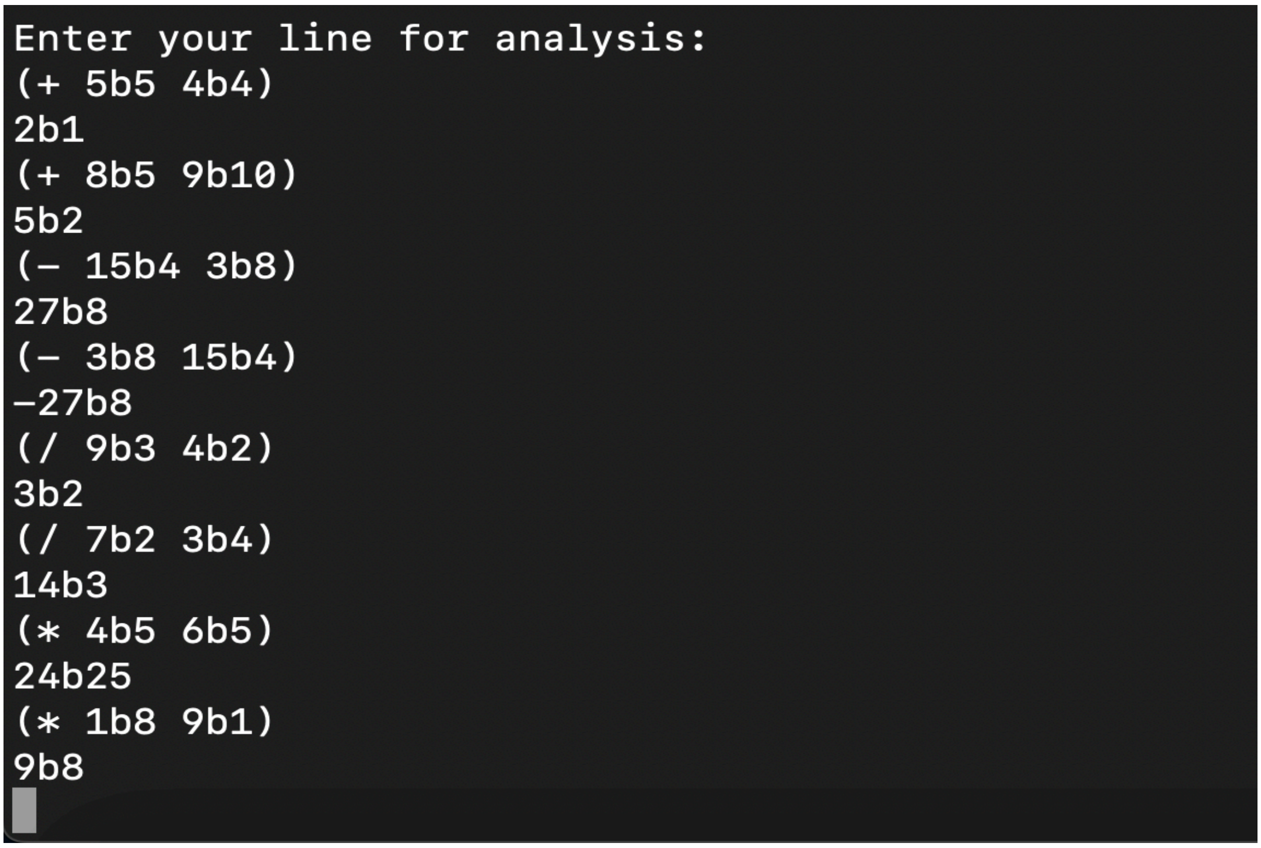
Documentation Of Homework3

Lisp Part

I created logical rules in Lisp language within the framework of the rules in the assignment and created both the lexical syntax control from the previous assignment and the concrete syntax given in this assignment.

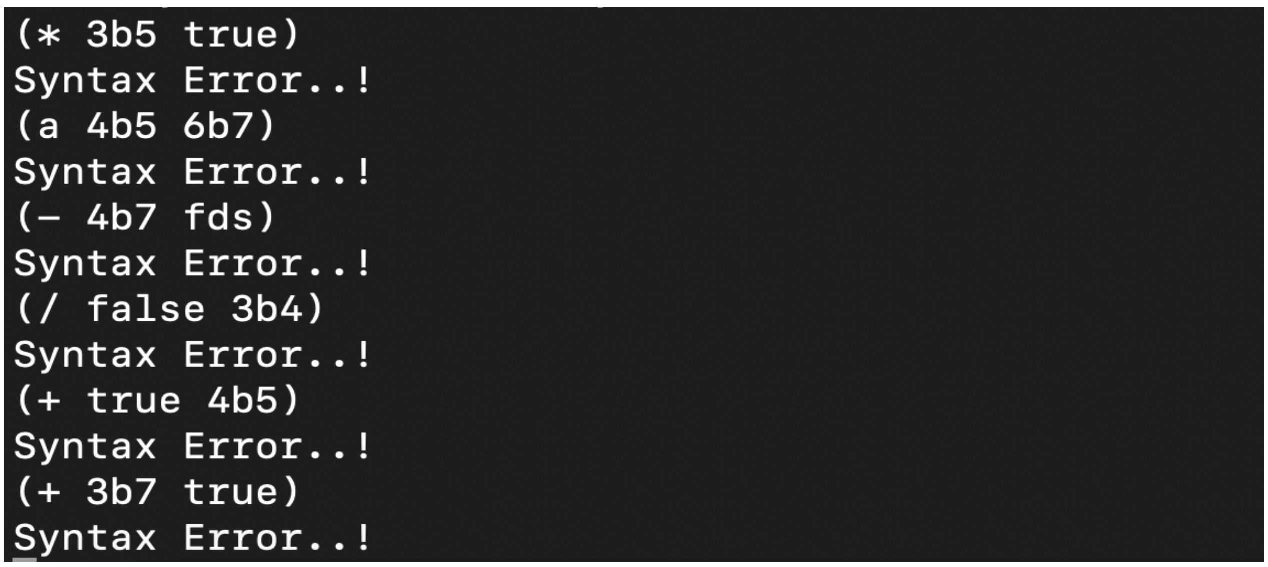
By creating these two syntaxes, I met the requirements of a language to a large extent and created a logical working structure. I directly warned the lexical mistakes made and caused errors and terminated the execution of the code, but I did not terminate the program in case of concrete syntax errors and I aimed to make many different attempts without terminating.

According to the concrete syntax I wrote, the words and or not equal less set def exit load display true false plus minus div mult valuef identifier exit can be used functionally.



```
Enter your line for analysis:
(+ 5b5 4b4)
2b1
(+ 8b5 9b10)
5b2
(- 15b4 3b8)
27b8
(- 3b8 15b4)
-27b8
(/ 9b3 4b2)
3b2
(/ 7b2 3b4)
14b3
(* 4b5 6b5)
24b25
(* 1b8 9b1)
9b8
```

Concrete syntax for operators: OP_OP OP_PLUS VALUEF VALUEF OP_CP


```
(* 3b5 true)
Syntax Error..!
(a 4b5 6b7)
Syntax Error..!
(- 4b7 fds)
Syntax Error..!
(/ false 3b4)
Syntax Error..!
(+ true 4b5)
Syntax Error..!
(+ 3b7 true)
Syntax Error..!
```

These are examples for error situations about operator

ERROR REASONS

1) 3b5 and true can not multiply

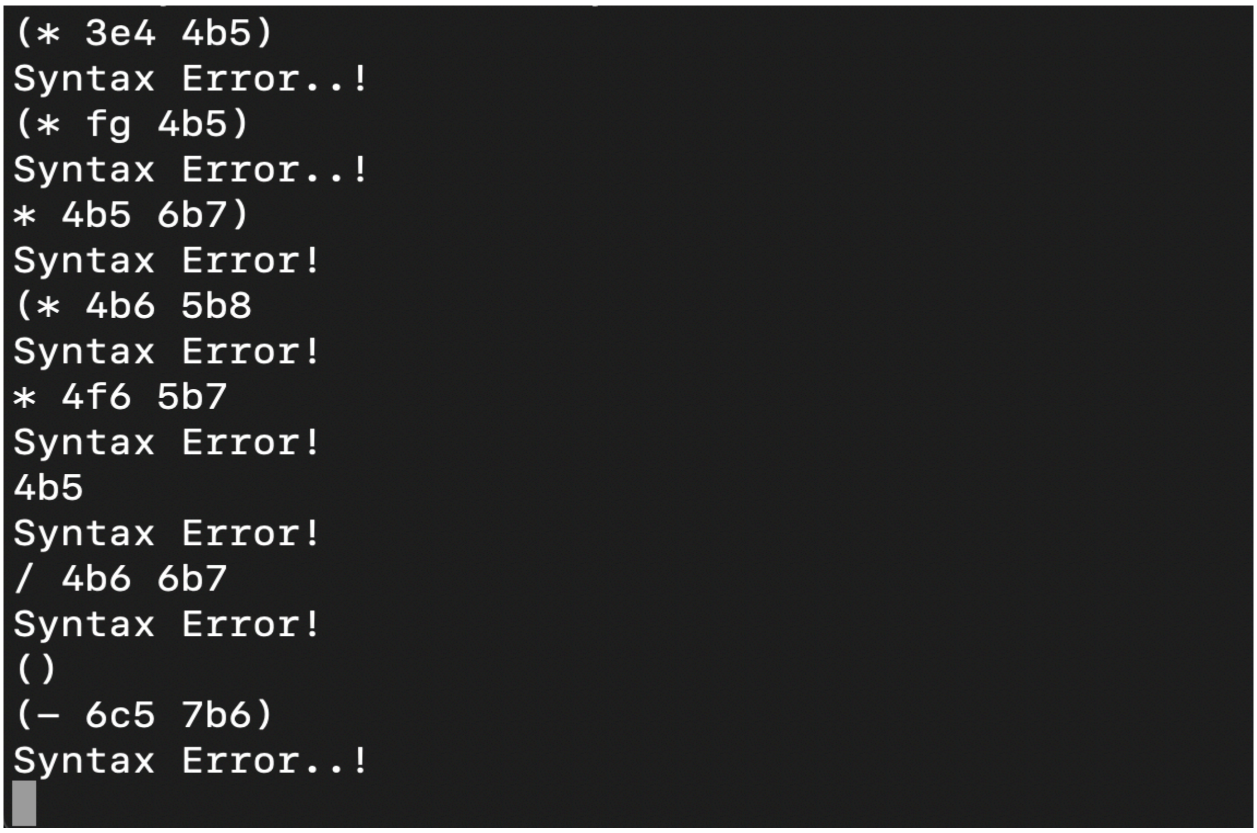
2) a is not operator

3) 4b7 and identifier can not subtract

4) false and 3b4 can not divide

5) true and 4b5 can not sum

6) 3b7 and true can not sum



```
(* 3e4 4b5)
Syntax Error..!
(* fg 4b5)
Syntax Error..!
* 4b5 6b7)
Syntax Error!
(* 4b6 5b8
Syntax Error!
* 4f6 5b7
Syntax Error!
4b5
Syntax Error!
/ 4b6 6b7
Syntax Error!
()
(- 6c5 7b6)
Syntax Error..!
```

These are general error situations for all concrete syntax

ERROR REASONS

1) 3e4 and 4b5 can not multiply

2) identifier and valuef can not be together in operator operations

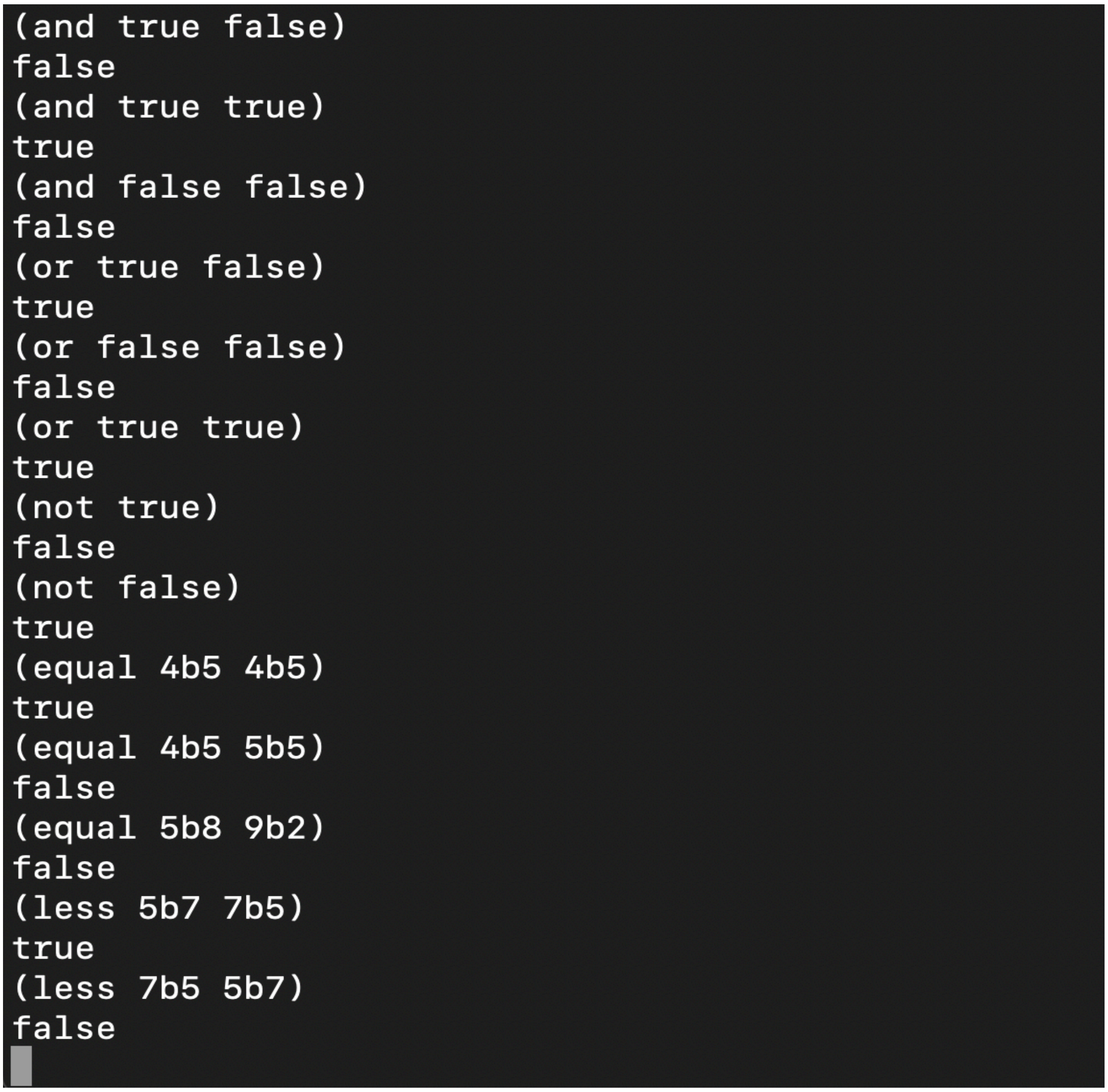
3) Must be paranthes in first

4) Must be paranthes in last

5) Must be paranthes in last and first

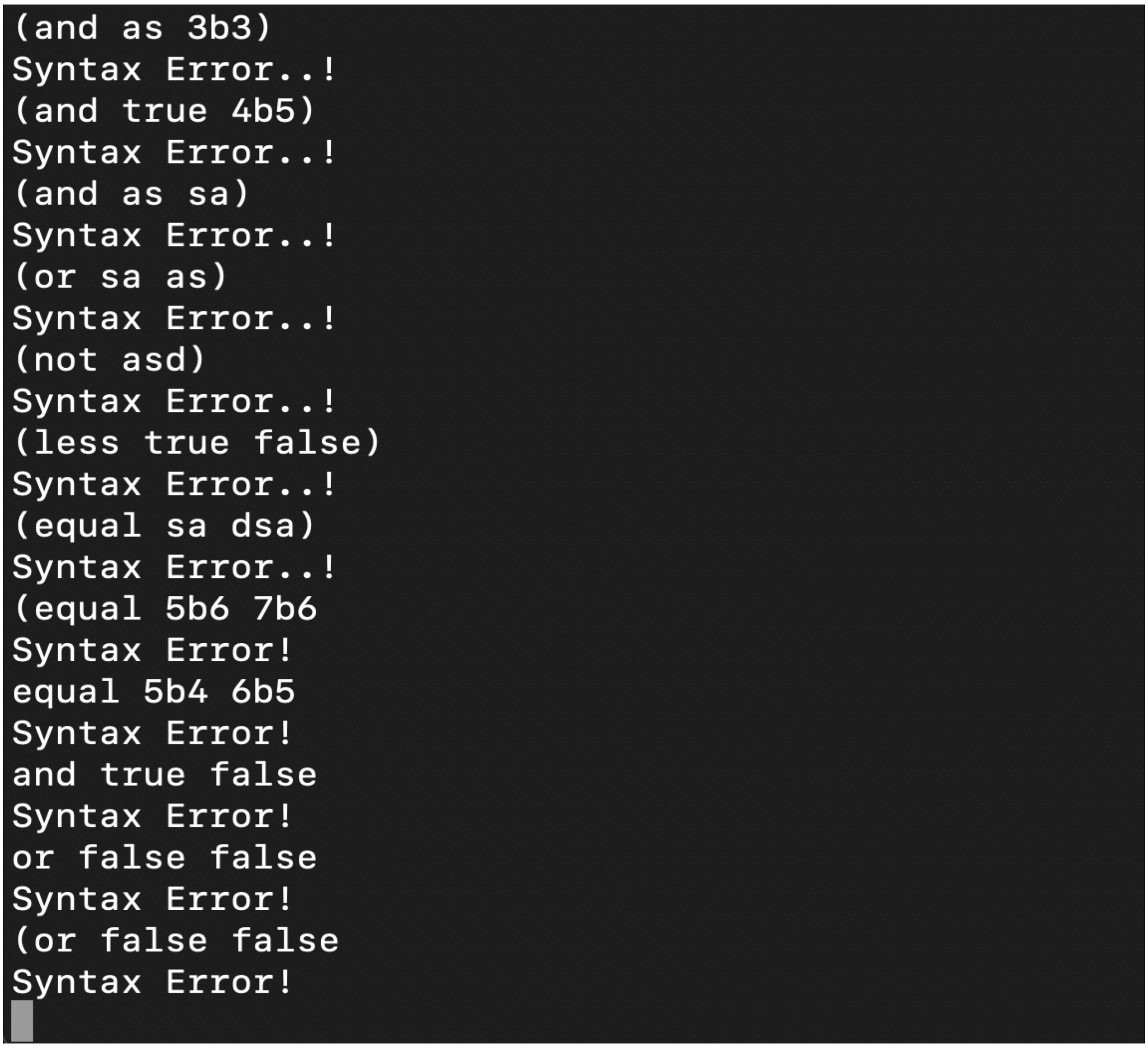
6) There is no meaning for just 4b5(valuef)

Others error situations same with aboves…



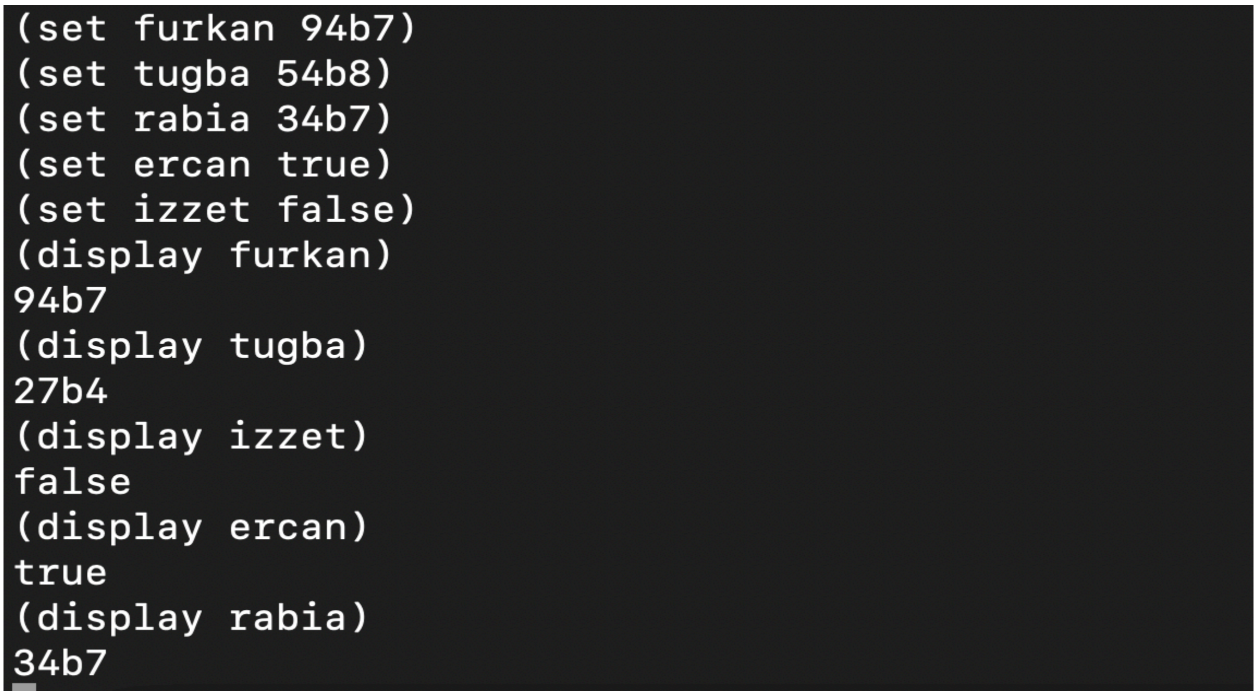
```
(and true false)
false
(and true true)
true
(and false false)
false
(or true false)
true
(or false false)
false
(or true true)
true
(not true)
false
(not false)
true
(equal 4b5 4b5)
true
(equal 4b5 5b5)
false
(equal 5b8 9b2)
false
(less 5b7 7b5)
true
(less 7b5 5b7)
false
```

Concrete syntax: OP_OP KW_AND true false OP_CP
Concrete syntax: OP_OP not true OP_CP



```
(and as 3b3)
Syntax Error..!
(and true 4b5)
Syntax Error..!
(and as sa)
Syntax Error..!
(or sa as)
Syntax Error..!
(not asd)
Syntax Error..!
(less true false)
Syntax Error..!
(equal sa dsa)
Syntax Error..!
(equal 5b6 7b6
Syntax Error!
equal 5b4 6b5
Syntax Error!
and true false
Syntax Error!
or false false
Syntax Error!
(or false false
Syntax Error!
```

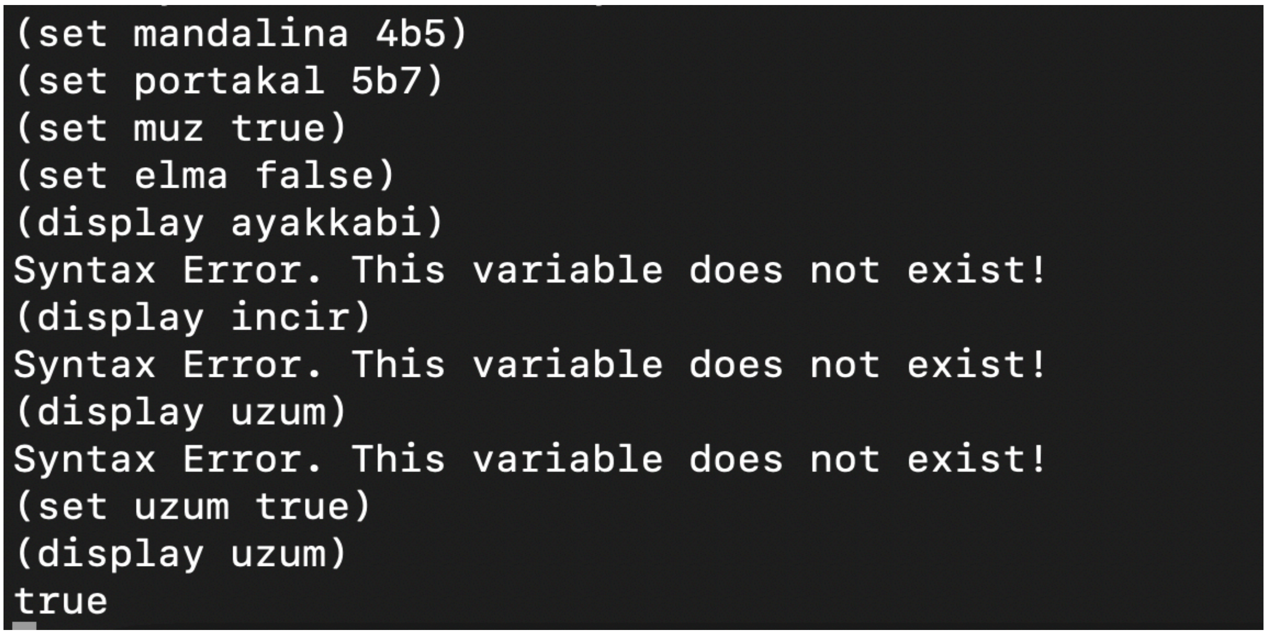
Since it did not comply with the concrete syntax I wrote, I printed the error on the screen in these cases.



```
(set furkan 94b7)
(set tugba 54b8)
(set rabia 34b7)
(set ercan true)
(set izzet false)
(display furkan)
94b7
(display tugba)
27b4
(display izzet)
false
(display ercan)
true
(display rabia)
34b7
```

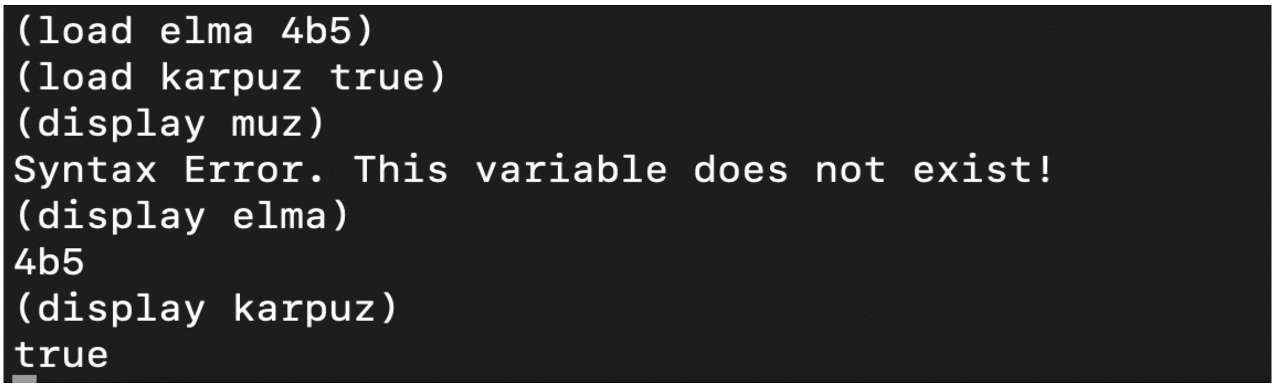
Concrete syntax: OP_OP set identifier valuef/true/false OP_CP ->set

Concrete syntax: OP_OP display identifier OP_CP ->display



```
(set mandalina 4b5)
(set portakal 5b7)
(set muz true)
(set elma false)
(display ayakkabi)
Syntax Error. This variable does not exist!
(display incir)
Syntax Error. This variable does not exist!
(display uzum)
Syntax Error. This variable does not exist!
(set uzum true)
(display uzum)
true
```

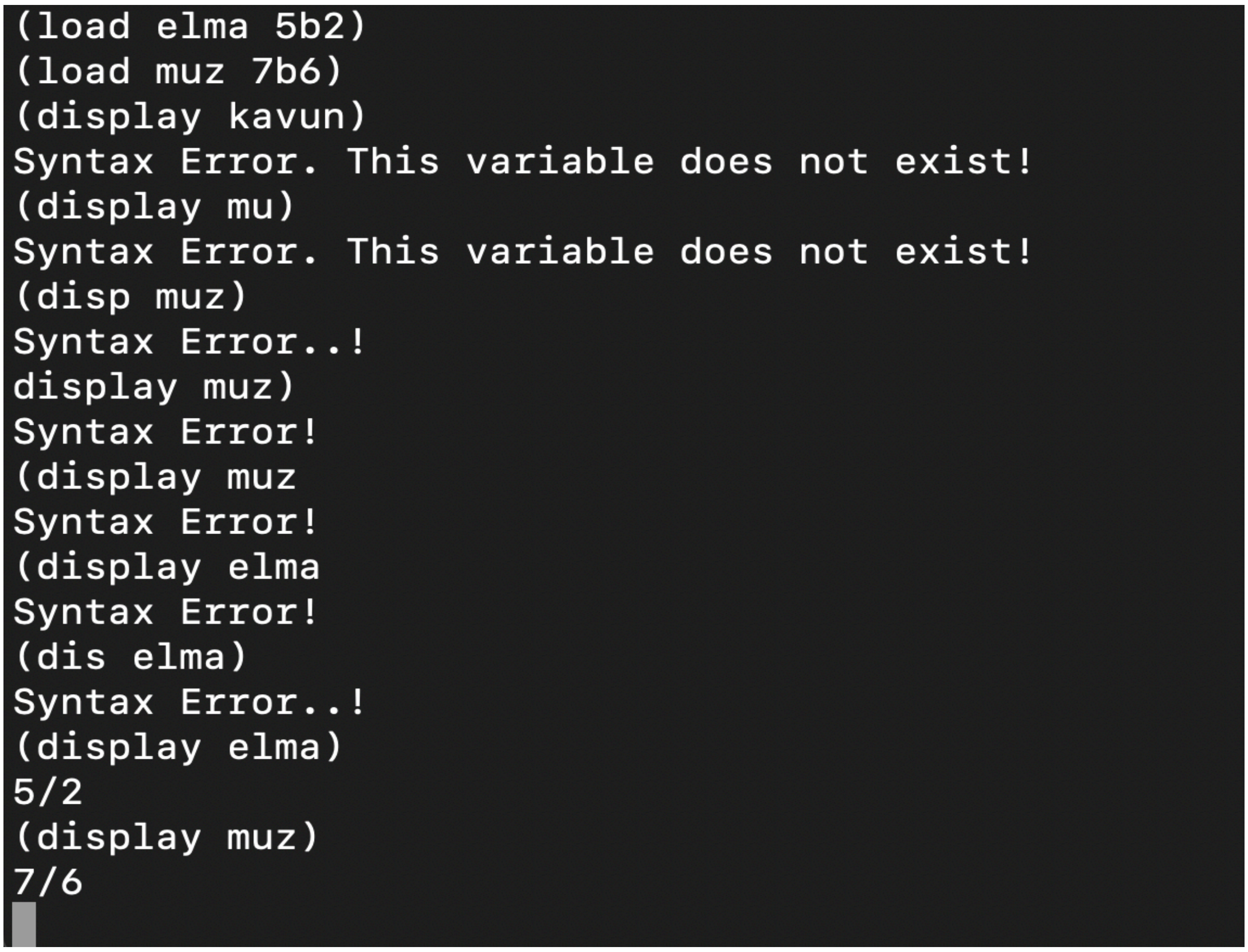
Since it did not comply with the concrete syntax I wrote, I printed the error on the screen in these cases.



```
(load elma 4b5)
(load karpuz true)
(display muz)
Syntax Error. This variable does not exist!
(display elma)
4b5
(display karpuz)
true
```

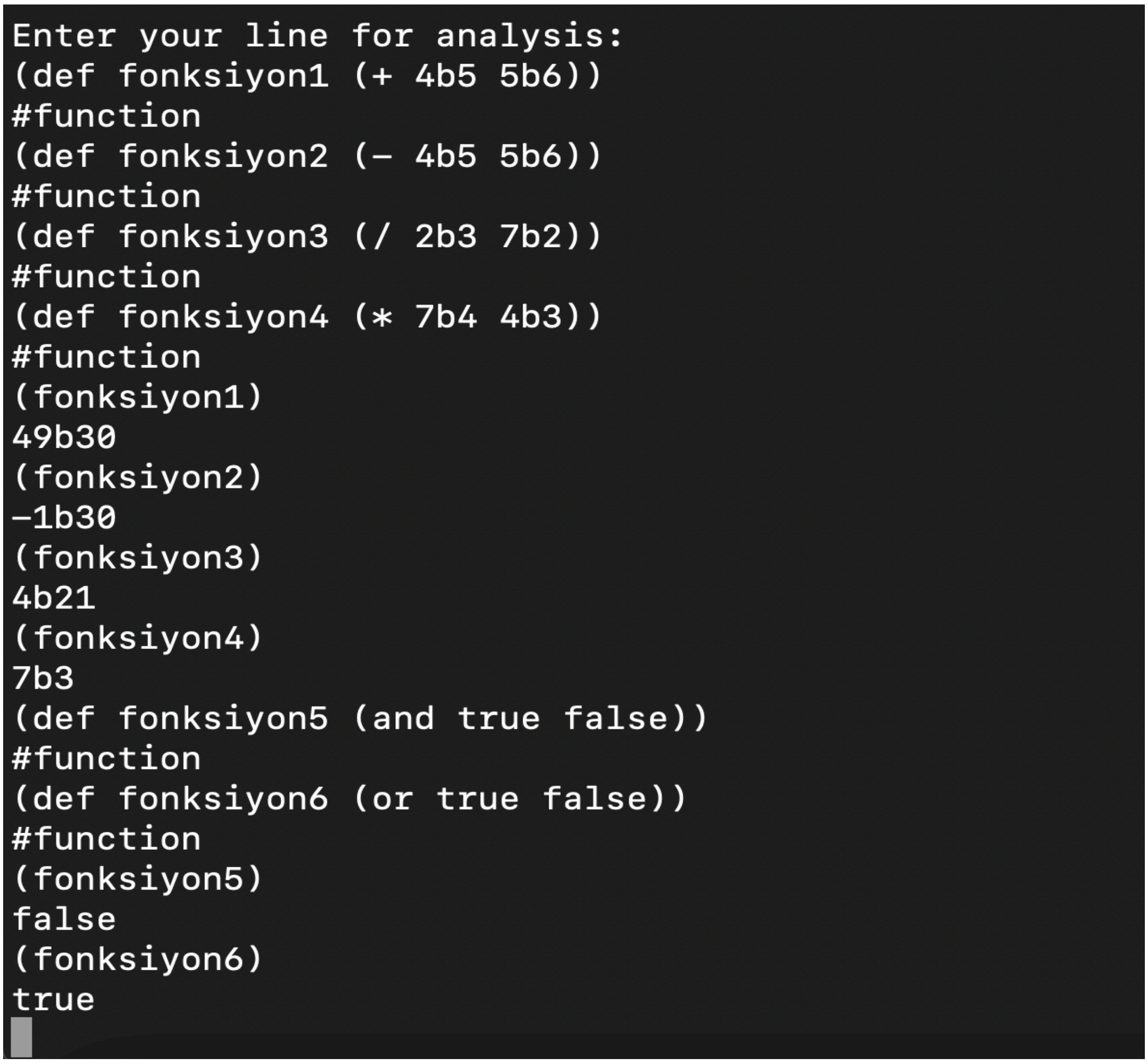
Examples of the use of the load keyword are given along with a possible error situation.

Concrete syntax: OP_OP load identifier valuef/true/false OP_CP ->load



```
(load elma 5b2)
(load muz 7b6)
(display kavun)
Syntax Error. This variable does not exist!
(display mu)
Syntax Error. This variable does not exist!
(disp muz)
Syntax Error..!
display muz)
Syntax Error!
(display muz
Syntax Error!
(display elma
Syntax Error!
(dis elma)
Syntax Error..!
(display elma)
5/2
(display muz)
7/6
```

Error situations are shown according to possible errors related to load usage and display.

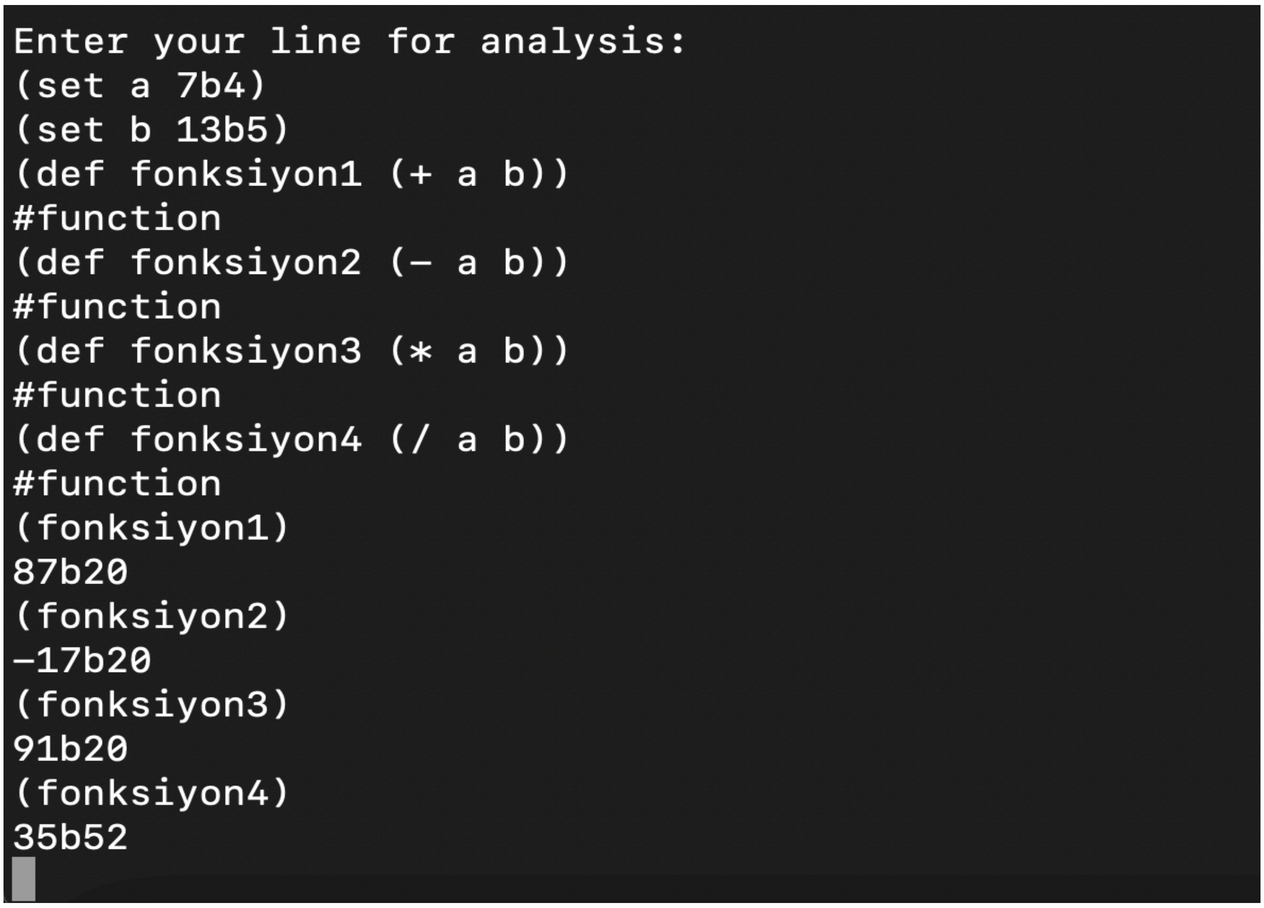


```
Enter your line for analysis:
(def fonksiyon1 (+ 4b5 5b6))
#function
(def fonksiyon2 (- 4b5 5b6))
#function
(def fonksiyon3 (/ 2b3 7b2))
#function
(def fonksiyon4 (* 7b4 4b3))
#function
(fonksiyon1)
49b30
(fonksiyon2)
-1b30
(fonksiyon3)
4b21
(fonksiyon4)
7b3
(def fonksiyon5 (and true false))
#function
(def fonksiyon6 (or true false))
#function
(fonksiyon5)
false
(fonksiyon6)
true
```

We can define functions using the def keyword and determine which numbers to operate in the functions, whether to use booleans or whether to use existing variables.

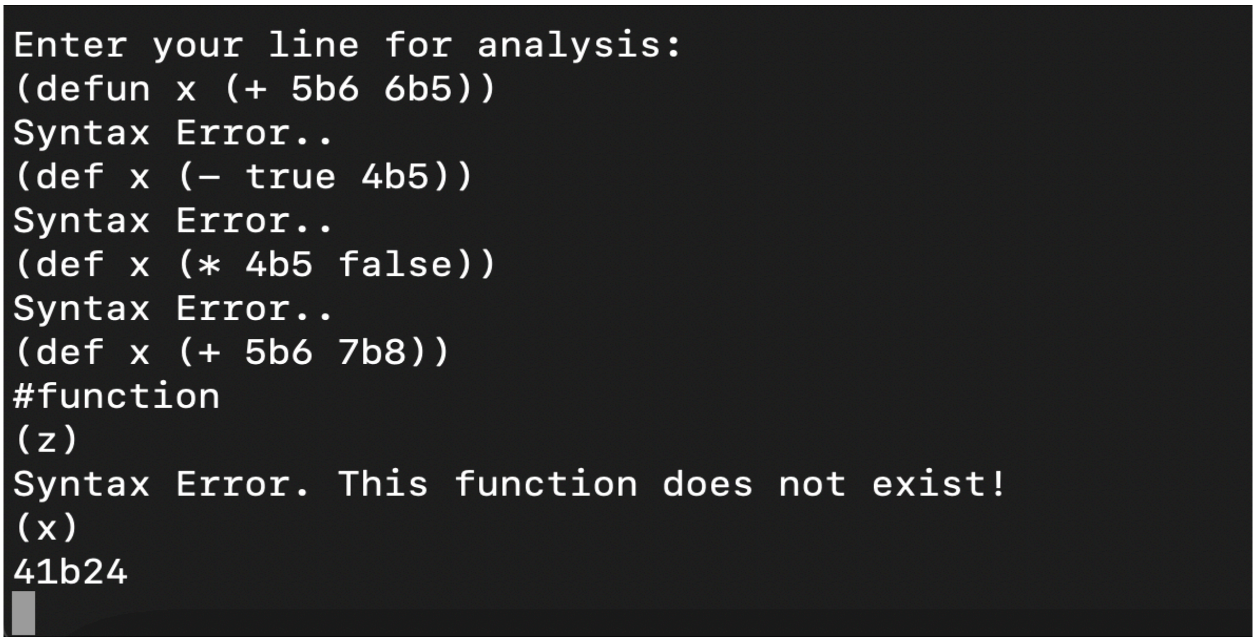
Concrete syntax: OP_OP def OP_OP operator valuef/identifier/booleans OP_CP OP_CP ->def

Concrete syntax: OP_OP identifier OP_CP ->function call



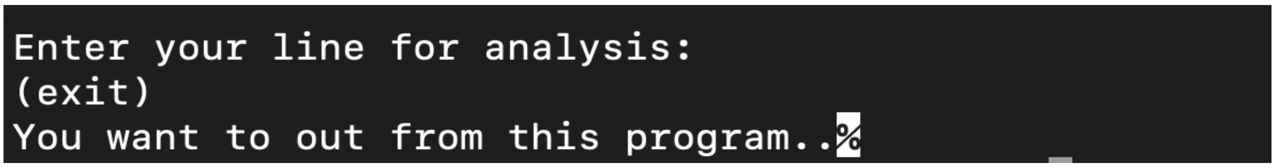
```
Enter your line for analysis:
(set a 7b4)
(set b 13b5)
(def fonksiyon1 (+ a b))
#function
(def fonksiyon2 (- a b))
#function
(def fonksiyon3 (* a b))
#function
(def fonksiyon4 (/ a b))
#function
(fonksiyon1)
87b20
(fonksiyon2)
-17b20
(fonksiyon3)
91b20
(fonksiyon4)
35b52
```

These are new example for function defining and calling



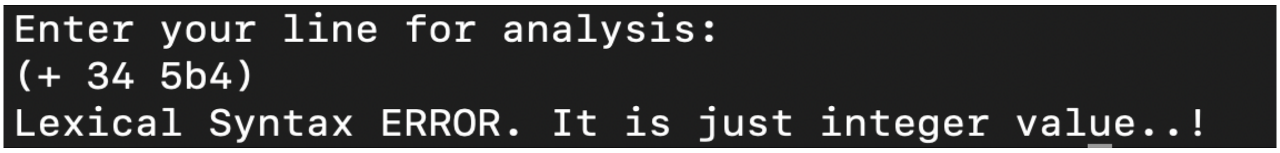
```
Enter your line for analysis:
(defun x (+ 5b6 6b5))
Syntax Error..
(def x (- true 4b5))
Syntax Error..
(def x (* 4b5 false))
Syntax Error..
(def x (+ 5b6 7b8))
#function
(z)
Syntax Error. This function does not exist!
(x)
41b24
```

These are example for situations of concrete errors

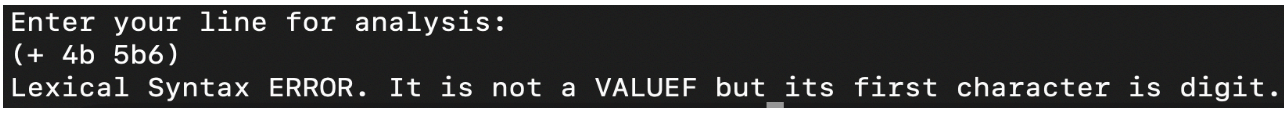


```
Enter your line for analysis:
(exit)
You want to out from this program..%
```

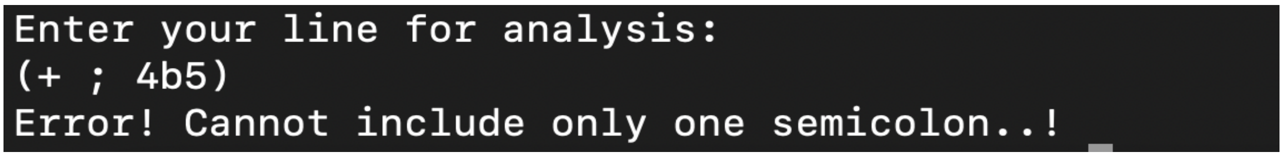
You can also out from this program as write (exit)



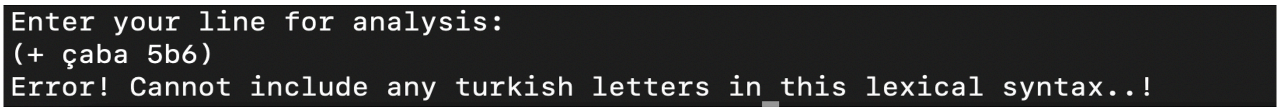
```
Enter your line for analysis:
(+ 34 5b4)
Lexical Syntax ERROR. It is just integer value..!
```



```
Enter your line for analysis:
(+ 4b 5b6)
Lexical Syntax ERROR. It is not a VALUEF but its first character is digit.
```

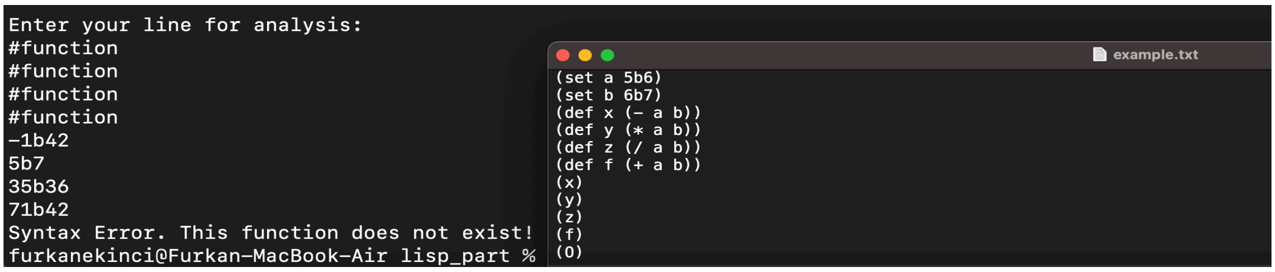


```
Enter your line for analysis:
(+ ; 4b5)
Error! Cannot include only one semicolon..!
```



```
Enter your line for analysis:
(+ çaba 5b6)
Error! Cannot include any turkish letters in this lexical syntax..!
```

These are example for lexical syntax errors



```
Enter your line for analysis:
#function
#function
#function
#function
-1b42
5b7
35b36
71b42
Syntax Error. This function does not exist!
furkanekinci@Furkan-MacBook-Air lisp_part %
```

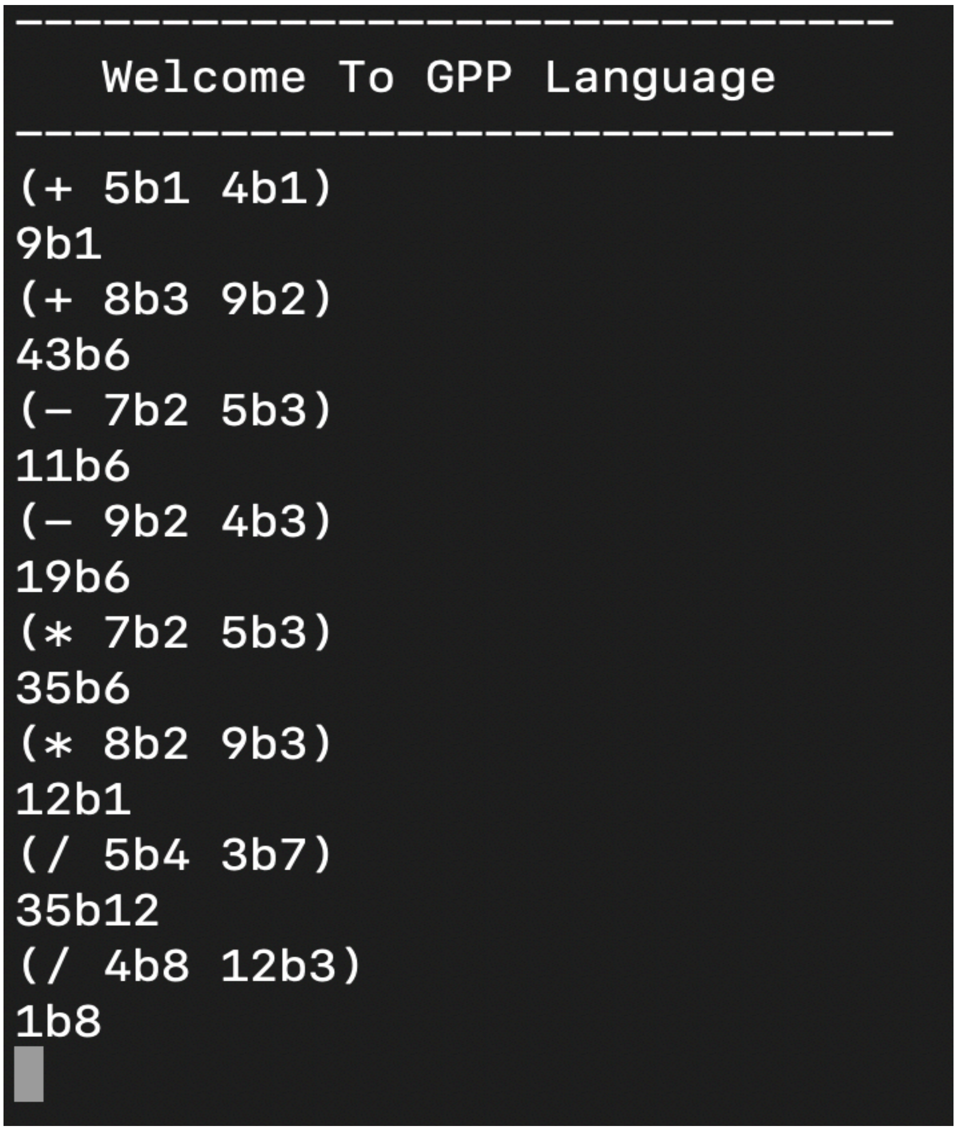
example.txt

```
(set a 5b6)
(set b 6b7)
(def x (- a b))
(def y (* a b))
(def z (/ a b))
(def f (+ a b))
(x)
(y)
(z)
(f)
(O)
```

I can also read what is written in the file and process it into the system, an example of which is given above. You can also try the code by making changes to the example.txt file located in the same directory as the project code. To run the code that only reads from the file, you must open the comment of the code in the bottom line of the lisp extension file and comment the next line.

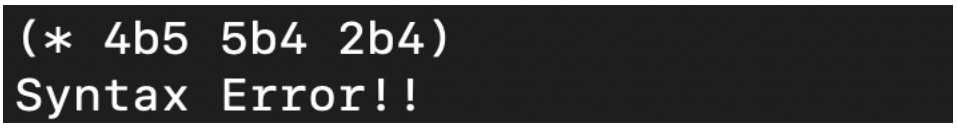
YACC Part

In this section, I determined all the situations to be exactly compatible with the concrete syntax given to us and created the concrete syntax of our language by writing special functions for all of them. I tried the examples for everything mentioned and in the examples in the pdf and I will share the outputs.



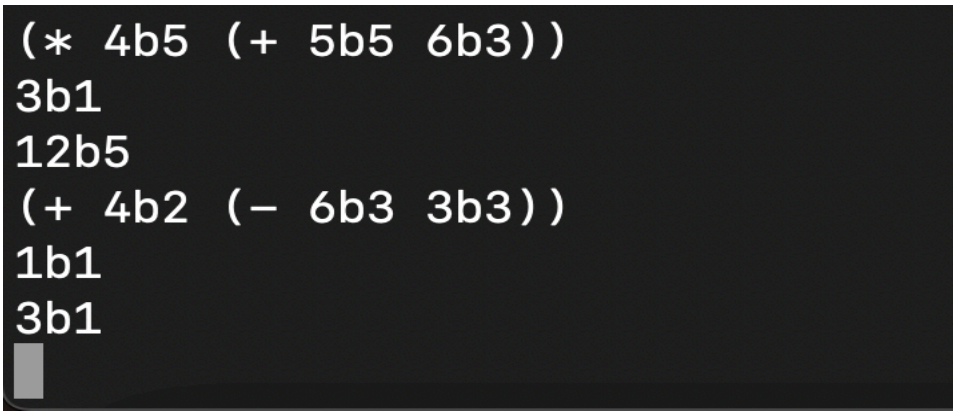
```
-------------------------------
   Welcome To GPP Language
-------------------------------
(+ 5b1 4b1)
9b1
(+ 8b3 9b2)
43b6
(- 7b2 5b3)
11b6
(- 9b2 4b3)
19b6
(* 7b2 5b3)
35b6
(* 8b2 9b3)
12b1
(/ 5b4 3b7)
35b12
(/ 4b8 12b3)
1b8
```

All 4 transactions are completed successfully



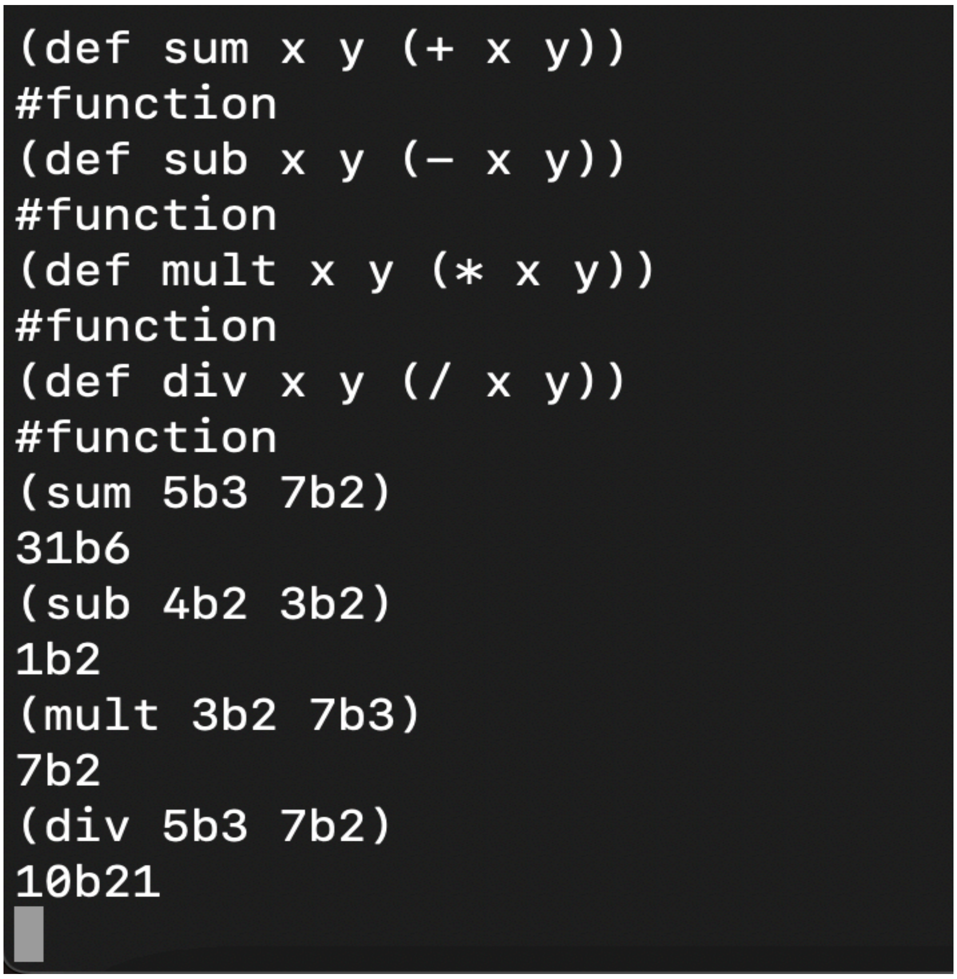
```
(* 4b5 5b4 2b4)
Syntax Error!!
```

Error status is shown for mathematical operations



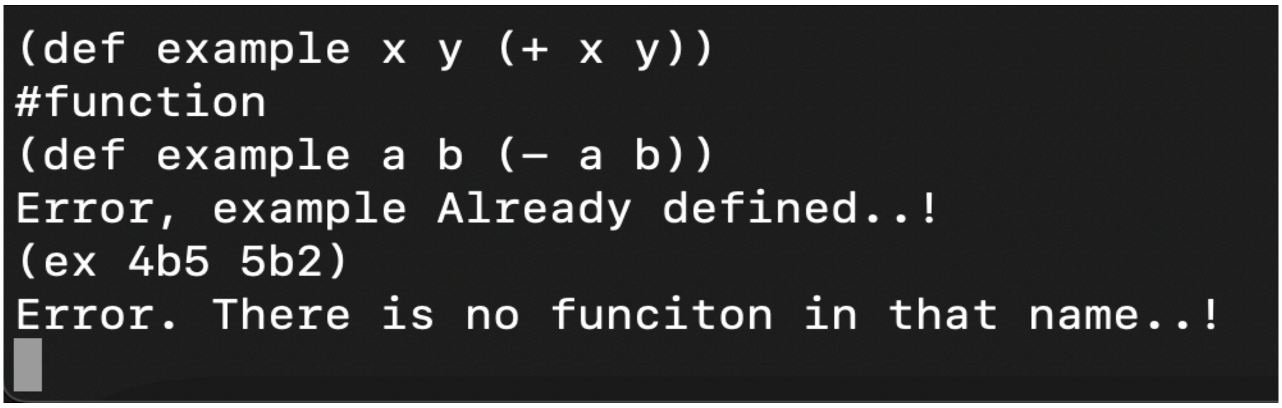
```
(* 4b5 (+ 5b5 6b3))
3b1
12b5
(+ 4b2 (- 6b3 3b3))
1b1
3b1
```

In nested mathematical operation calls, I first printed the result of the innermost one and then the last main result on the screen.



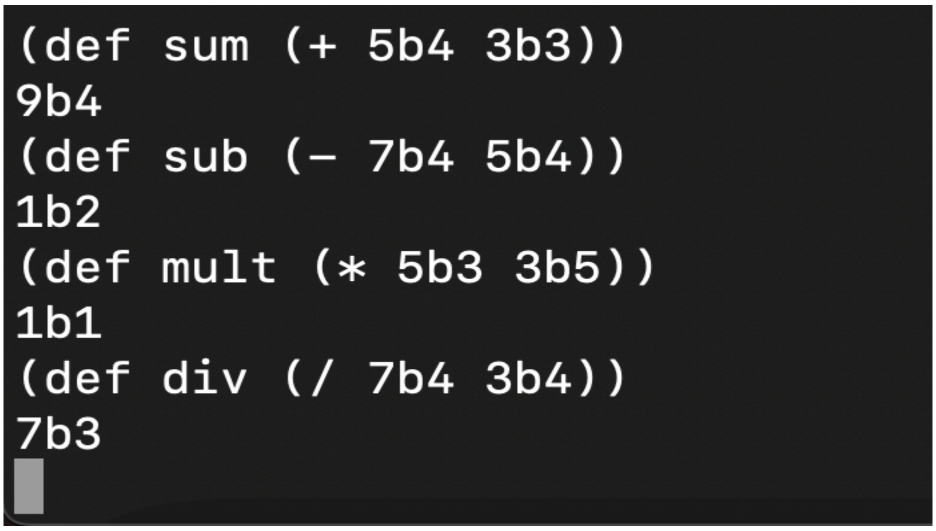
```
(def sum x y (+ x y))
#function
(def sub x y (- x y))
#function
(def mult x y (* x y))
#function
(def div x y (/ x y))
#function
(sum 5b3 7b2)
31b6
(sub 4b2 3b2)
1b2
(mult 3b2 7b3)
7b2
(div 5b3 7b2)
10b21
```

I defined and called function for each operator with two parameters



```
(def example x y (+ x y))
#function
(def example a b (- a b))
Error, example Already defined..!
(ex 4b5 5b2)
Error. There is no funciton in that name..!
```

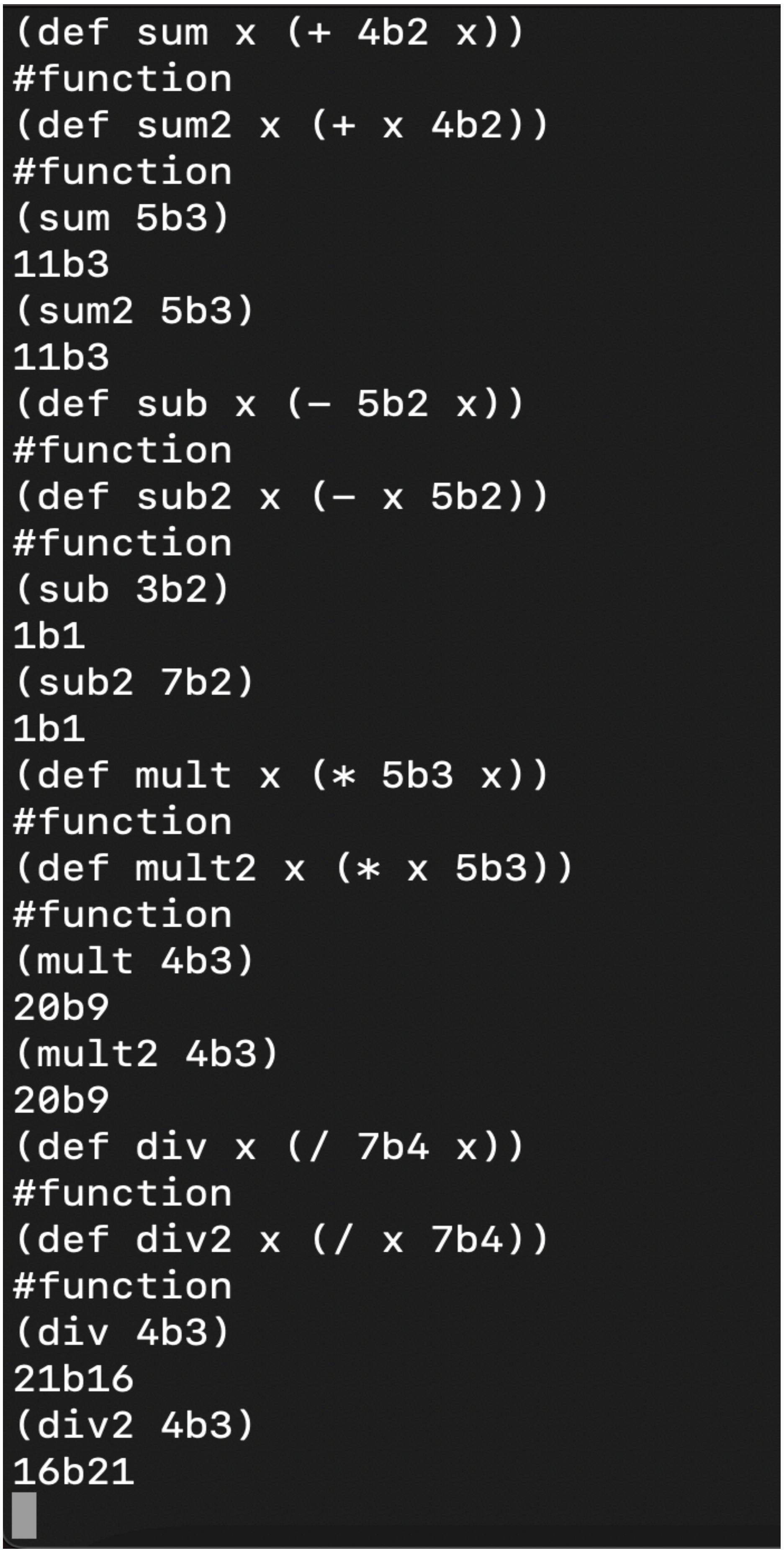
When you try to use a previously defined function name, it gives an error. Additionally, when you try to run a function that has not yet been defined, a special error is displayed on the screen.



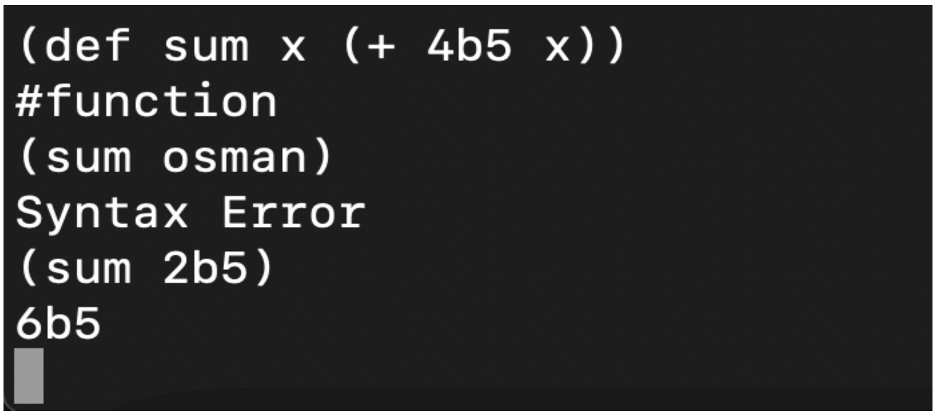
```
(def sum (+ 5b4 3b3))
9b4
(def sub (- 7b4 5b4))
1b2
(def mult (* 5b3 3b5))
1b1
(def div (/ 7b4 3b4))
7b3
```

I created functions without parameters and printed their results to the screen

Below, I created single-parameter functions and showed both ways to show that single-parameter functions can be run when the parameter is both on the right and left.

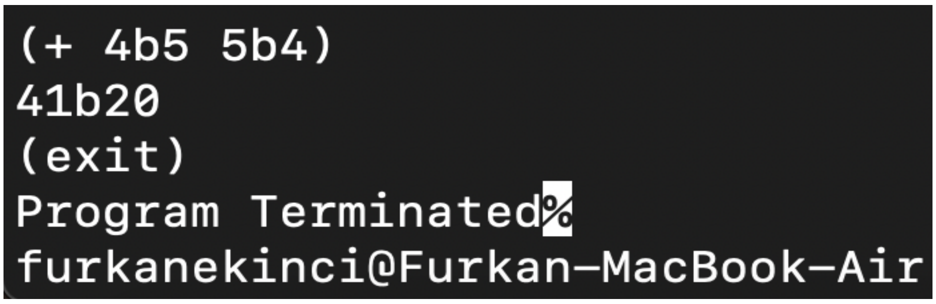


```
(def sum x (+ 4b2 x))
#function
(def sum2 x (+ x 4b2))
#function
(sum 5b3)
11b3
(sum2 5b3)
11b3
(def sub x (– 5b2 x))
#function
(def sub2 x (– x 5b2))
#function
(sub 3b2)
1b1
(sub2 7b2)
1b1
(def mult x (* 5b3 x))
#function
(def mult2 x (* x 5b3))
#function
(mult 4b3)
20b9
(mult2 4b3)
20b9
(def div x (/ 7b4 x))
#function
(def div2 x (/ x 7b4))
#function
(div 4b3)
21b16
(div2 4b3)
16b21
```



```
(def sum x (+ 4b5 x))
#function
(sum osman)
Syntax Error
(sum 2b5)
6b5
```

When a function is called with a name that has not been created, it gives an error because that function has not been created.



```
(+ 4b5 5b4)
41b20
(exit)
Program Terminated%
furkanekinci@Furkan-MacBook-Air
```

To exit the program, simply type "(exit)"

NOTE: You can run the lisp and yacc parts with the make command.

Ahmet Furkan Ekinci
"200104004063"