Programming Languages
Homework 3 Report

Mustafa Tokgöz

171044077

Flex and Yacc)

In this part , if command line argument exists read input from file otherwise read input from terminal until user enter (exit).

Usage:

```
make
./a.out filename or
./a.out
```

In this part I use my previous lex file for flex. In yacc file, I use struct for identifiers to keep their values as table. Also I use global array for list values and printing them.Then I write the code with tokens and I implement them to show results.The programs runs when the user enter (exit) with respect to homework pdf. In argument file also must include (exit) text to exit the program.When the user enter wrong input program ends also.In flex and yacc part, programs works fine.

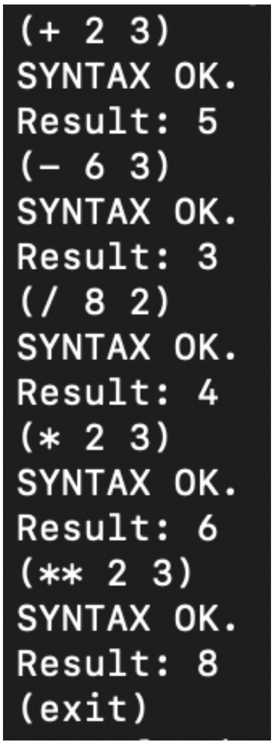
Implemented properties : Everything except deffun command

Unimplemented properties: deffun command

I explain how I handle them upper side.

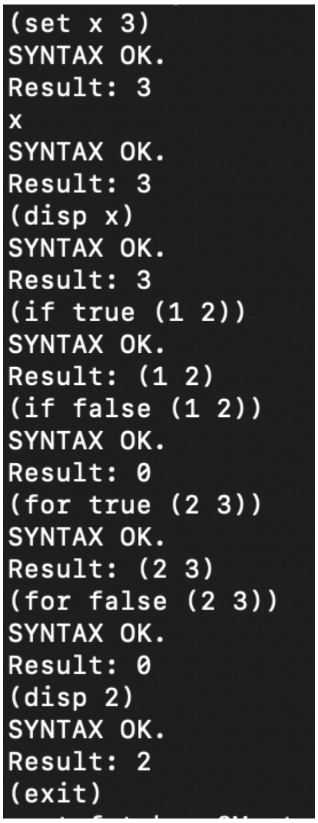
Ouputs:

For this output , this output shows the mathematic operations in expi.



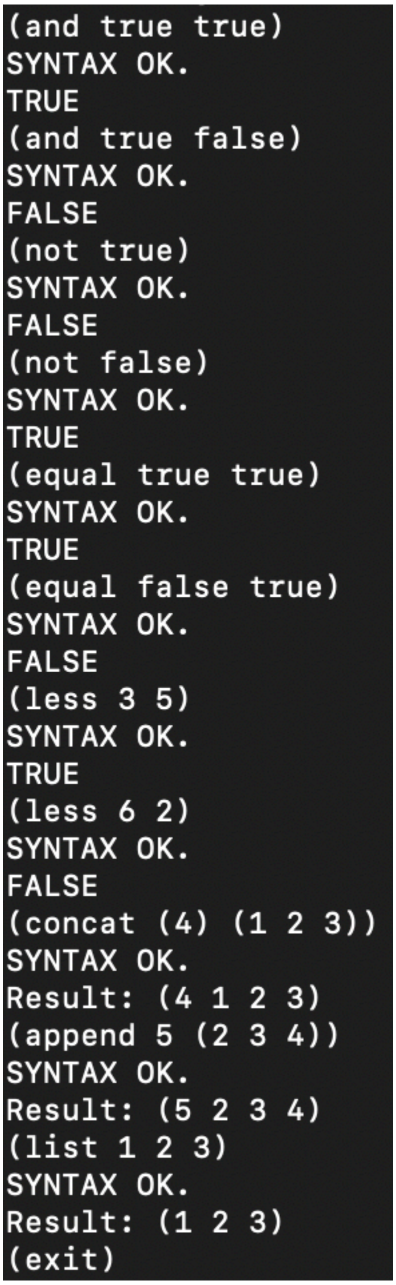
```
(+ 2 3)
SYNTAX OK.
Result: 5
(- 6 3)
SYNTAX OK.
Result: 3
(/ 8 2)
SYNTAX OK.
Result: 4
(* 2 3)
SYNTAX OK.
Result: 6
(** 2 3)
SYNTAX OK.
Result: 8
(exit)
```

For this output, I showed set,if,for and disp commands' outputs in expi.

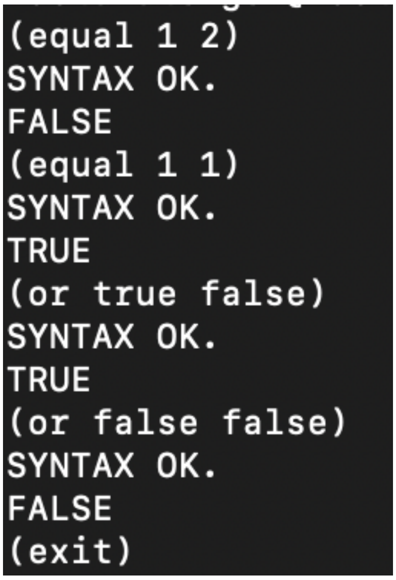


```
(set x 3)
SYNTAX OK.
Result: 3
x
SYNTAX OK.
Result: 3
(disp x)
SYNTAX OK.
Result: 3
(if true (1 2))
SYNTAX OK.
Result: (1 2)
(if false (1 2))
SYNTAX OK.
Result: 0
(for true (2 3))
SYNTAX OK.
Result: (2 3)
(for false (2 3))
SYNTAX OK.
Result: 0
(disp 2)
SYNTAX OK.
Result: 2
(exit)
```

For this output, I showed and,not,equal,less,or commands’ outputs in expb.



```
(and true true)
SYNTAX OK.
TRUE
(and true false)
SYNTAX OK.
FALSE
(not true)
SYNTAX OK.
FALSE
(not false)
SYNTAX OK.
TRUE
(equal true true)
SYNTAX OK.
TRUE
(equal false true)
SYNTAX OK.
FALSE
(less 3 5)
SYNTAX OK.
TRUE
(less 6 2)
SYNTAX OK.
FALSE
(concat (4) (1 2 3))
SYNTAX OK.
Result: (4 1 2 3)
(append 5 (2 3 4))
SYNTAX OK.
Result: (5 2 3 4)
(list 1 2 3)
SYNTAX OK.
Result: (1 2 3)
(exit)
```



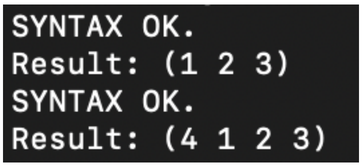
```
(equal 1 2)
SYNTAX OK.
FALSE
(equal 1 1)
SYNTAX OK.
TRUE
(or true false)
SYNTAX OK.
TRUE
(or false false)
SYNTAX OK.
FALSE
(exit)
```

From the file

File text :

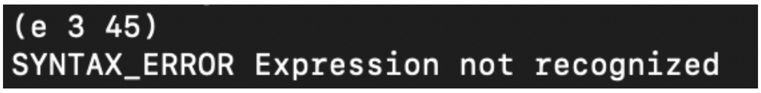
```
(list 1 2 3)
(append 4 (1 2 3))
(exit)
```

Output :



```
SYNTAX OK.
Result: (1 2 3)
SYNTAX OK.
Result: (4 1 2 3)
```

For the wrong input:



```
(e 3 45)
SYNTAX_ERROR Expression not recognized
```