

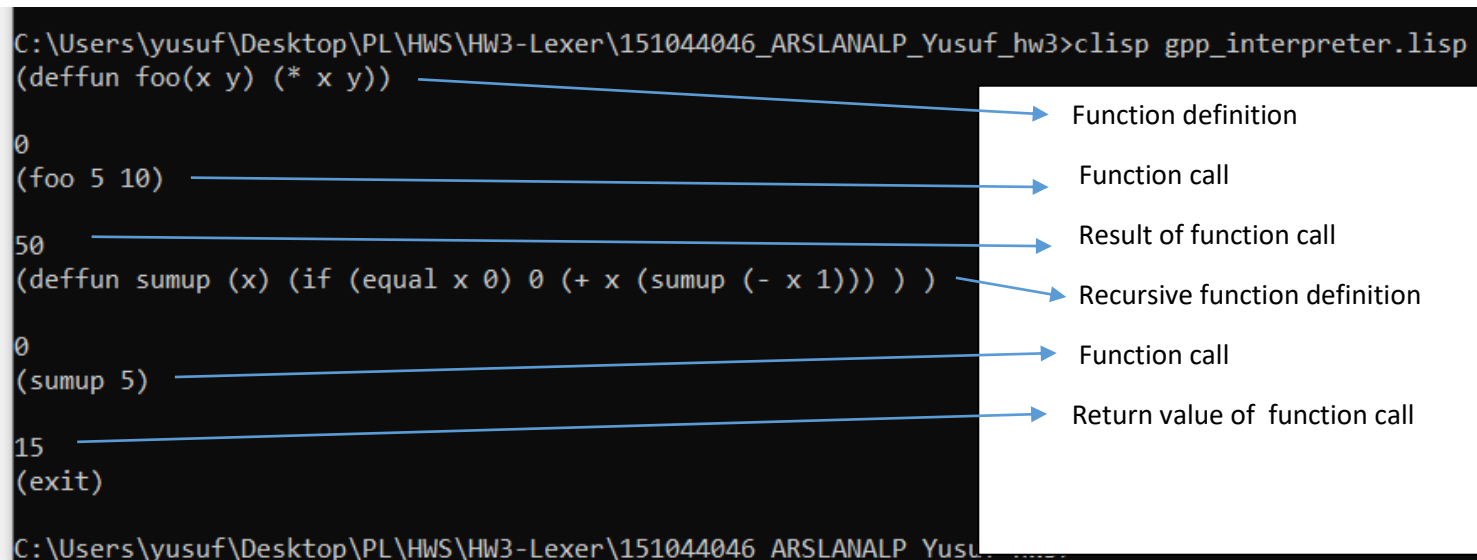
Programming Languages HW3 Report

What I Implemented:

- I didn't Implement yacc part.
- I implemented Lisp part.

What I Implemented in lisp part:

- Function definition (include recursive)
- Function call
- Assignment (defvar and set functions)
- If statement
- Arithmetic operators (+, -, /, *)
- Boolean operators (and, or)
- Equal function (equal expi expi)



The screenshot shows a terminal window with the following text:

```
C:\Users\yusuf\Desktop\PL\HWS\HW3-Lexer\151044046_ARSLANALP_Yusuf_hw3>clisp gpp_interpreter.lisp
(deffun foo(x y) (* x y))
0
(foo 5 10)
50
(deffun sumup (x) (if (equal x 0) 0 (+ x (sumup (- x 1))) ) )
0
(sumup 5)
15
(exit)
C:\Users\yusuf\Desktop\PL\HWS\HW3-Lexer\151044046_ARSLANALP_Yusuf_hw3>
```

Annotations on the right side of the terminal output:

- Function definition (points to `(deffun foo(x y) (* x y))`)
- Function call (points to `(foo 5 10)`)
- Result of function call (points to `50`)
- Recursive function definition (points to `(deffun sumup (x) (if (equal x 0) 0 (+ x (sumup (- x 1)))))`)
- Function call (points to `(sumup 5)`)
- Return value of function call (points to `15`)

There is a Sample output in above. You can see from there that I implemented function definition and function call. Even recursive function definition and function calls work.

NOTE: Ignore zeros in above screenshots. They are just return values of function definition. According to assignment PDF, function definition should return zero.

How to run program:

- `clisp gpp_interpreter.lisp`
- `clisp gpp_interpreter.lisp input.txt`

Yusuf Abdullah ARSLANALP

151044046