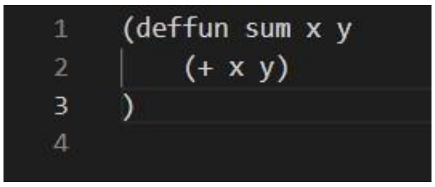
Burak Kocausta

1901042605

CSE 341 – HW2

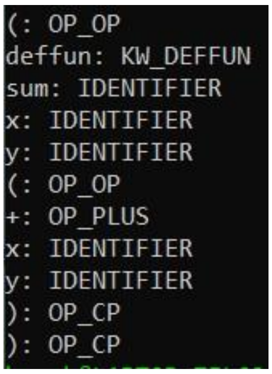
PART 1: FLEX

Case – input.txt:



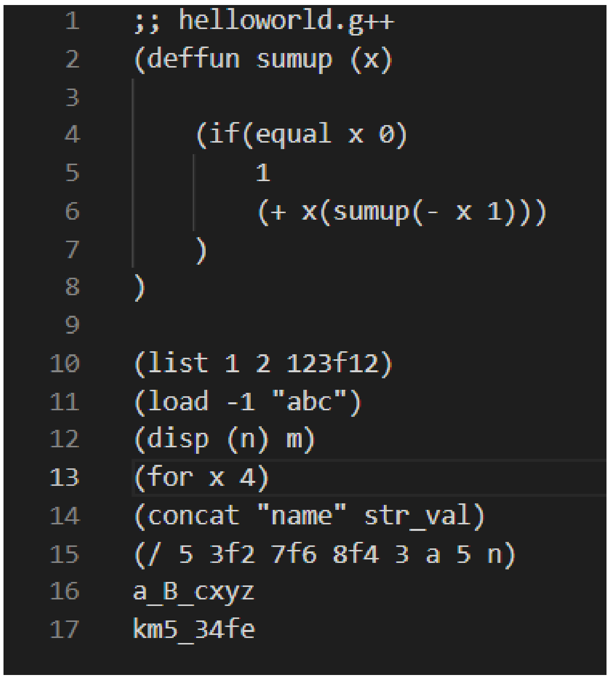
```
(deffun sum x y
    (+ x y)
)

```

Result – input.txt:



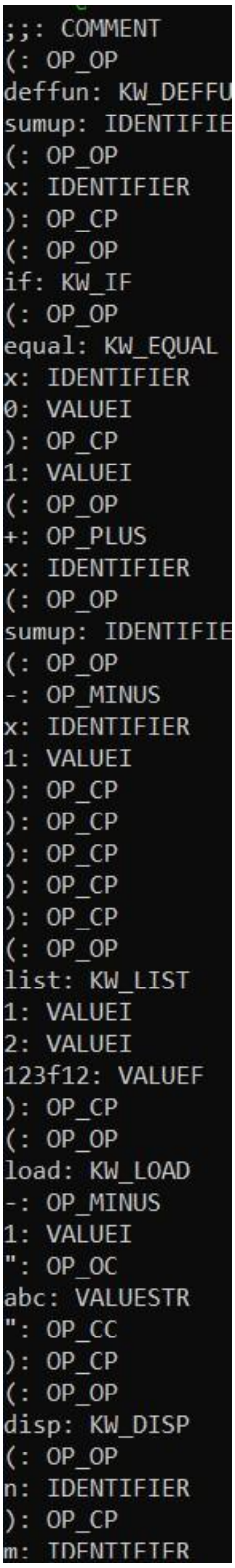
```
(: OP_OP
deffun: KW_DEFFUN
sum: IDENTIFIER
x: IDENTIFIER
y: IDENTIFIER
(: OP_OP
+: OP_PLUS
x: IDENTIFIER
y: IDENTIFIER
): OP_CP
): OP_CP
```

Case – test1.g++

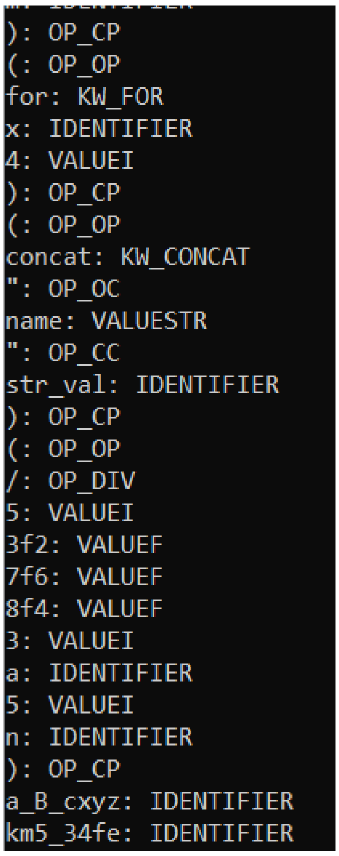


```
;; helloworld.g++
(deffun sumup (x)

    (if(equal x 0)
        1
        (+ x(sumup(- x 1)))
    )
)

(list 1 2 123f12)
(load -1 "abc")
(disp (n) m)
(for x 4)
(concat "name" str_val)
(/ 5 3f2 7f6 8f4 3 a 5 n)
a_B_cxyz
km5_34fe
```

Result – test1.g++

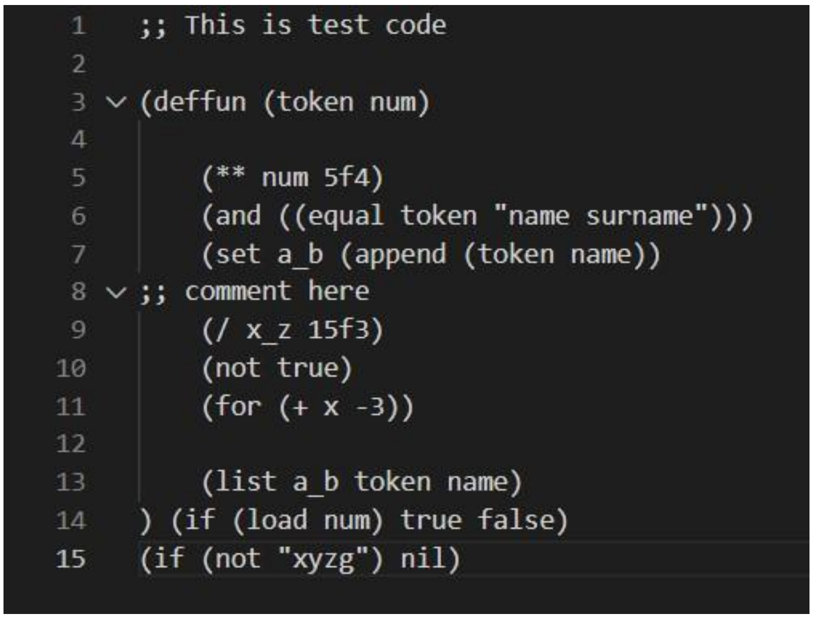


```
;;: COMMENT
(: OP_OP
deffun: KW_DEFFU
sumup: IDENTIFIE
(: OP_OP
x: IDENTIFIER
): OP_CP
(: OP_OP
if: KW_IF
(: OP_OP
equal: KW_EQUAL
x: IDENTIFIER
0: VALUEI
): OP_CP
1: VALUEI
(: OP_OP
+: OP_PLUS
x: IDENTIFIER
(: OP_OP
sumup: IDENTIFIE
(: OP_OP
-: OP_MINUS
x: IDENTIFIER
1: VALUEI
): OP_CP
): OP_CP
): OP_CP
): OP_CP
): OP_CP
(: OP_OP
list: KW_LIST
1: VALUEI
2: VALUEI
123f12: VALUEF
): OP_CP
(: OP_OP
load: KW_LOAD
-: OP_MINUS
1: VALUEI
": OP_OC
abc: VALUESTR
": OP_CC
): OP_CP
(: OP_OP
disp: KW_DISP
(: OP_OP
n: IDENTIFIER
): OP_CP
m: IDENTIFIER
```



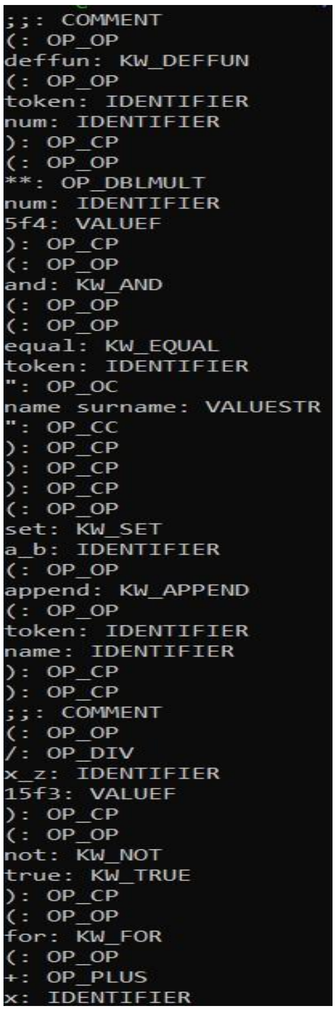
```
): OP_CP
(: OP_OP
for: KW_FOR
x: IDENTIFIER
4: VALUEI
): OP_CP
(: OP_OP
concat: KW_CONCAT
": OP_OC
name: VALUESTR
": OP_CC
str_val: IDENTIFIER
): OP_CP
(: OP_OP
/: OP_DIV
5: VALUEI
3f2: VALUEF
7f6: VALUEF
8f4: VALUEF
3: VALUEI
a: IDENTIFIER
5: VALUEI
n: IDENTIFIER
): OP_CP
a_B_cxyz: IDENTIFIER
km5_34fe: IDENTIFIER
```

Case – test2.g++

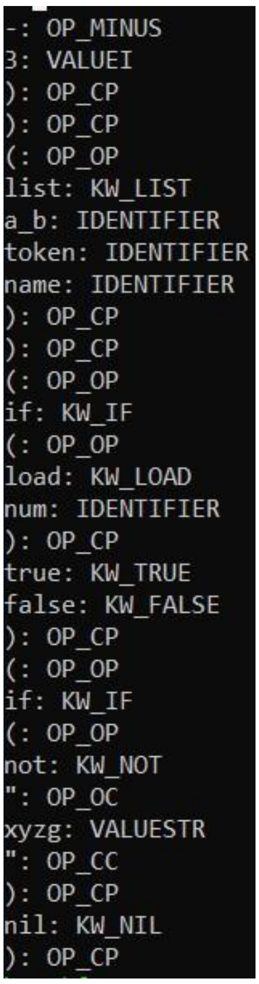


```
;; This is test code

(deffun (token num)

    (** num 5f4)
    (and ((equal token "name surname")))
    (set a_b (append (token name))
;; comment here
    (/ x_z 15f3)
    (not true)
    (for (+ x -3))

    (list a_b token name)
) (if (load num) true false)
(if (not "xyzg") nil)
```

Result – test2.g++

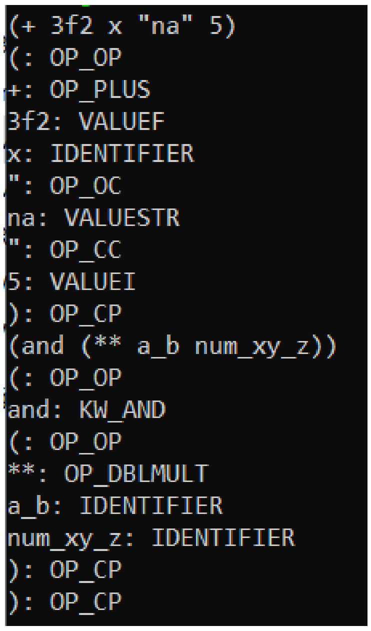


```
;;: COMMENT
(: OP_OP
deffun: KW_DEFFUN
(: OP_OP
token: IDENTIFIER
num: IDENTIFIER
): OP_CP
(: OP_OP
**: OP_DBLMULT
num: IDENTIFIER
5f4: VALUEF
): OP_CP
(: OP_OP
and: KW_AND
(: OP_OP
(: OP_OP
equal: KW_EQUAL
token: IDENTIFIER
": OP_OC
name surname: VALUESTR
": OP_CC
): OP_CP
): OP_CP
): OP_CP
(: OP_OP
set: KW_SET
a_b: IDENTIFIER
(: OP_OP
append: KW_APPEND
(: OP_OP
token: IDENTIFIER
name: IDENTIFIER
): OP_CP
): OP_CP
;;: COMMENT
(: OP_OP
/: OP_DIV
x_z: IDENTIFIER
15f3: VALUEF
): OP_CP
(: OP_OP
not: KW_NOT
true: KW_TRUE
): OP_CP
(: OP_OP
for: KW_FOR
(: OP_OP
+: OP_PLUS
x: IDENTIFIER
```



```
-: OP_MINUS
3: VALUEI
): OP_CP
): OP_CP
(: OP_OP
list: KW_LIST
a_b: IDENTIFIER
token: IDENTIFIER
name: IDENTIFIER
): OP_CP
): OP_CP
(: OP_OP
if: KW_IF
(: OP_OP
load: KW_LOAD
num: IDENTIFIER
): OP_CP
true: KW_TRUE
false: KW_FALSE
): OP_CP
(: OP_OP
if: KW_IF
(: OP_OP
not: KW_NOT
": OP_OC
xyzg: VALUESTR
": OP_CC
): OP_CP
nil: KW_NIL
): OP_CP
```

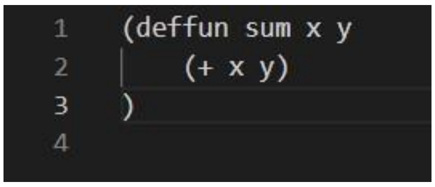
REPL TEST



```
(+ 3f2 x "na" 5)
(: OP_OP
+: OP_PLUS
3f2: VALUEF
x: IDENTIFIER
": OP_OC
na: VALUESTR
": OP_CC
5: VALUEI
): OP_CP
(and (** a_b num_xy_z))
(: OP_OP
and: KW_AND
(: OP_OP
**: OP_DBLMULT
a_b: IDENTIFIER
num_xy_z: IDENTIFIER
): OP_CP
): OP_CP
```

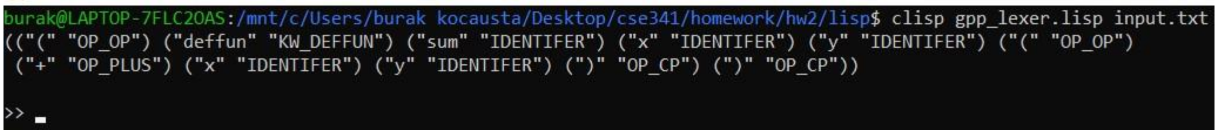
PART 2: LISP

Case – input.txt:



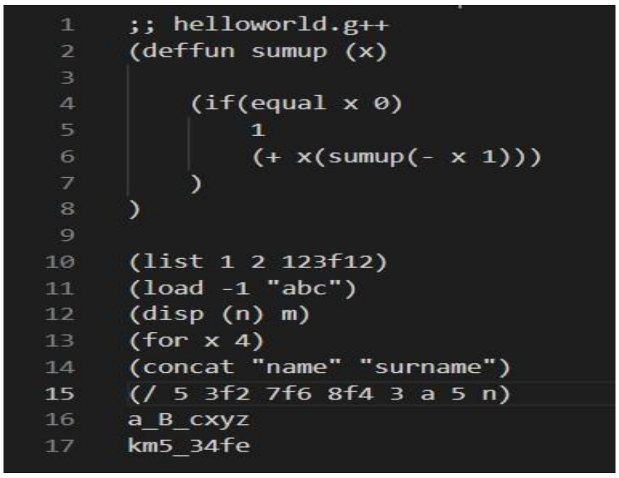
```
(deffun sum x y
    (+ x y)
)

```

Result – input.txt:



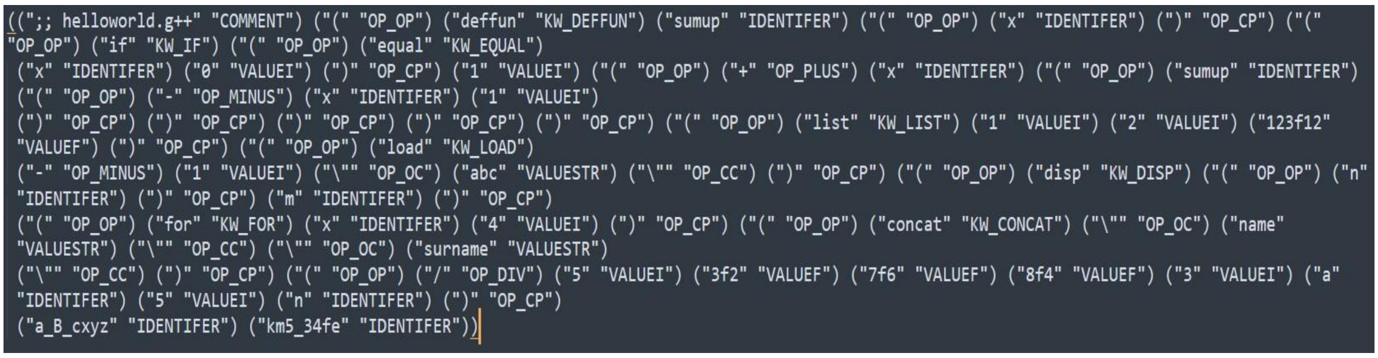
```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse341/homework/hw2/lisp$ clisp gpp_lexer.lisp input.txt
(("(" "OP_OP") ("deffun" "KW_DEFFUN") ("sum" "IDENTIFER") ("x" "IDENTIFER") ("y" "IDENTIFER") ("(" "OP_OP")
 ("+" "OP_PLUS") ("x" "IDENTIFER") ("y" "IDENTIFER") (")" "OP_CP") (")" "OP_CP"))

>> _
```

Case – test1.g++:



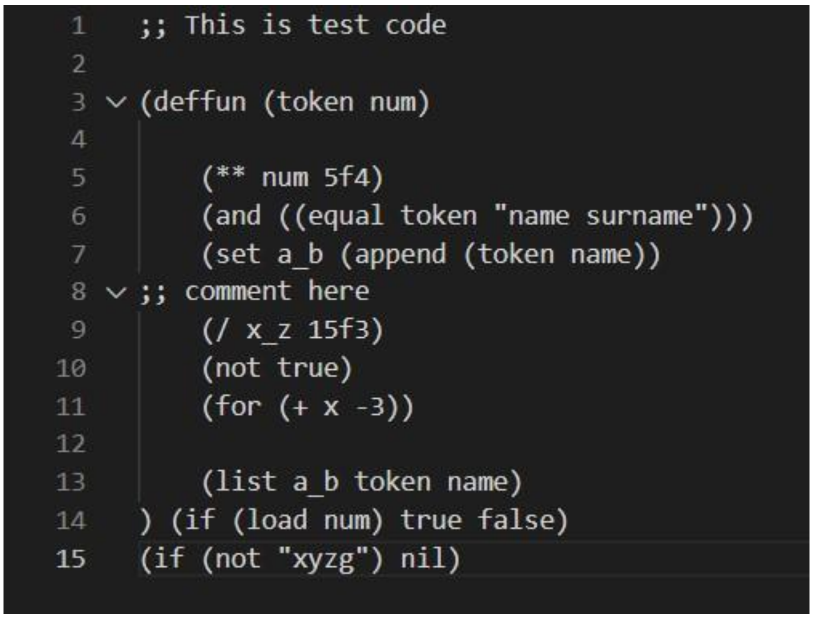
```
;; helloworld.g++
(deffun sumup (x)

    (if(equal x 0)
        1
        (+ x(sumup(- x 1)))
    )
)

(list 1 2 123f12)
(load -1 "abc")
(disp (n) m)
(for x 4)
(concat "name" "surname")
(/ 5 3f2 7f6 8f4 3 a 5 n)
a_B_cxyz
km5_34fe
```

Result – test1.g++:



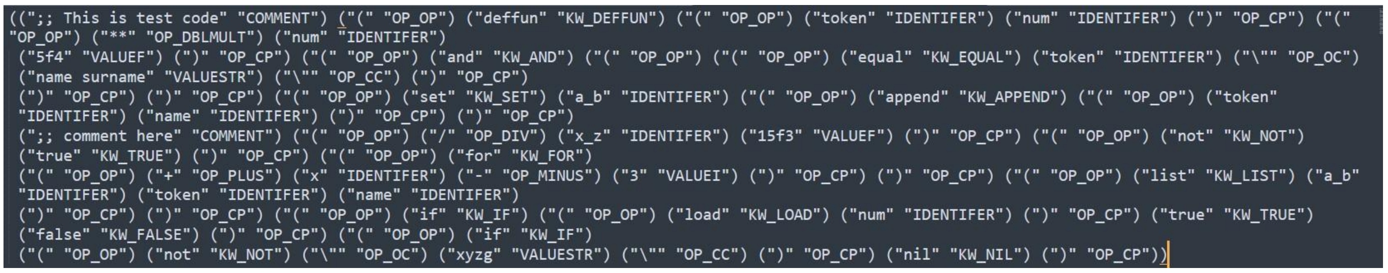
```
((";; helloworld.g++" "COMMENT") ("(" "OP_OP") ("deffun" "KW_DEFFUN") ("sumup" "IDENTIFER") ("(" "OP_OP") ("x" "IDENTIFER") (")" "OP_CP") ("("
"OP_OP") ("if" "KW_IF") ("(" "OP_OP") ("equal" "KW_EQUAL")
 ("x" "IDENTIFER") ("0" "VALUEI") (")" "OP_CP") ("1" "VALUEI") ("(" "OP_OP") ("+" "OP_PLUS") ("x" "IDENTIFER") ("(" "OP_OP") ("sumup" "IDENTIFER")
 ("(" "OP_OP") ("-" "OP_MINUS") ("x" "IDENTIFER") ("1" "VALUEI")
 (")" "OP_CP") (")" "OP_CP") (")" "OP_CP") (")" "OP_CP") (")" "OP_CP") ("(" "OP_OP") ("list" "KW_LIST") ("1" "VALUEI") ("2" "VALUEI") ("123f12"
 "VALUEF") (")" "OP_CP") ("(" "OP_OP") ("load" "KW_LOAD")
 ("-" "OP_MINUS") ("1" "VALUEI") ("\"" "OP_OC") ("abc" "VALUESTR") ("\"" "OP_CC") (")" "OP_CP") ("(" "OP_OP") ("disp" "KW_DISP") ("(" "OP_OP") ("n"
 "IDENTIFER") (")" "OP_CP") ("m" "IDENTIFER") (")" "OP_CP")
 ("(" "OP_OP") ("for" "KW_FOR") ("x" "IDENTIFER") ("4" "VALUEI") (")" "OP_CP") ("(" "OP_OP") ("concat" "KW_CONCAT") ("\"" "OP_OC") ("name"
 "VALUESTR") ("\"" "OP_CC") ("\"" "OP_OC") ("surname" "VALUESTR")
 ("\"" "OP_CC") (")" "OP_CP") ("(" "OP_OP") ("/" "OP_DIV") ("5" "VALUEI") ("3f2" "VALUEF") ("7f6" "VALUEF") ("8f4" "VALUEF") ("3" "VALUEI") ("a"
 "IDENTIFER") ("5" "VALUEI") ("n" "IDENTIFER") (")" "OP_CP")
 ("a_B_cxyz" "IDENTIFER") ("km5_34fe" "IDENTIFER"))
```

Case – test2.g++:



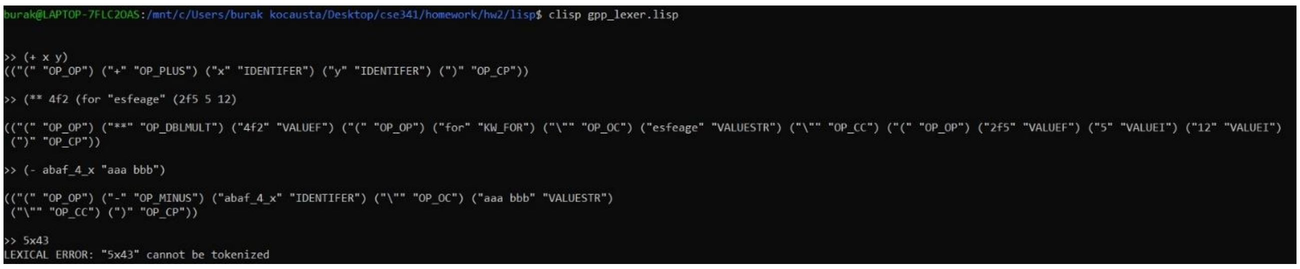
```
;; This is test code

(deffun (token num)

    (** num 5f4)
    (and ((equal token "name surname")))
    (set a_b (append (token name))
;; comment here
    (/ x_z 15f3)
    (not true)
    (for (+ x -3))

    (list a_b token name)
) (if (load num) true false)
(if (not "xyzg") nil)
```

Result – test2.g++:



```
((";; This is test code" "COMMENT") ("(" "OP_OP") ("deffun" "KW_DEFFUN") ("(" "OP_OP") ("token" "IDENTIFER") ("num" "IDENTIFER") (")" "OP_CP") ("("
"OP_OP") ("**" "OP_DBLMULT") ("num" "IDENTIFER")
 ("5f4" "VALUEF") (")" "OP_CP") ("(" "OP_OP") ("and" "KW_AND") ("(" "OP_OP") ("(" "OP_OP") ("equal" "KW_EQUAL") ("token" "IDENTIFER") ("\"" "OP_OC")
 ("name surname" "VALUESTR") ("\"" "OP_CC") (")" "OP_CP")
 (")" "OP_CP") (")" "OP_CP") ("(" "OP_OP") ("set" "KW_SET") ("a_b" "IDENTIFER") ("(" "OP_OP") ("append" "KW_APPEND") ("(" "OP_OP") ("token"
 "IDENTIFER") ("name" "IDENTIFER") (")" "OP_CP") (")" "OP_CP")
 (";; comment here" "COMMENT") ("(" "OP_OP") ("/" "OP_DIV") ("x_z" "IDENTIFER") ("15f3" "VALUEF") (")" "OP_CP") ("(" "OP_OP") ("not" "KW_NOT")
 ("true" "KW_TRUE") (")" "OP_CP") ("(" "OP_OP") ("for" "KW_FOR")
 ("(" "OP_OP") ("+" "OP_PLUS") ("x" "IDENTIFER") ("-" "OP_MINUS") ("3" "VALUEI") (")" "OP_CP") (")" "OP_CP") ("(" "OP_OP") ("list" "KW_LIST") ("a_b"
 "IDENTIFER") ("token" "IDENTIFER") ("name" "IDENTIFER")
 (")" "OP_CP") (")" "OP_CP") ("(" "OP_OP") ("if" "KW_IF") ("(" "OP_OP") ("load" "KW_LOAD") ("num" "IDENTIFER") (")" "OP_CP") ("true" "KW_TRUE")
 ("false" "KW_FALSE") (")" "OP_CP") ("(" "OP_OP") ("if" "KW_IF")
 ("(" "OP_OP") ("not" "KW_NOT") ("\"" "OP_OC") ("xyzg" "VALUESTR") ("\"" "OP_CC") (")" "OP_CP") ("nil" "KW_NIL") (")" "OP_CP"))
```

REPL TEST



```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse341/homework/hw2/lisp$ clisp gpp_lexer.lisp


>> (+ x y)
(("(" "OP_OP") ("+" "OP_PLUS") ("x" "IDENTIFER") ("y" "IDENTIFER") (")" "OP_CP"))

>> (** 4f2 (for "esfeage" (2f5 5 12)

(("(" "OP_OP") ("**" "OP_DBLMULT") ("4f2" "VALUEF") ("(" "OP_OP") ("for" "KW_FOR") ("\"" "OP_OC") ("esfeage" "VALUESTR") ("\"" "OP_CC") ("(" "OP_OP") ("2f5" "VALUEF") ("5" "VALUEI") ("12" "VALUEI")
 (")" "OP_CP"))

>> (- abaf_4_x "aaa bbb")

(("(" "OP_OP") ("-" "OP_MINUS") ("abaf_4_x" "IDENTIFER") ("\"" "OP_OC") ("aaa bbb" "VALUESTR")
 ("\"" "OP_CC") (")" "OP_CP"))

>> 5x43
LEXICAL ERROR: "5x43" cannot be tokenized
```