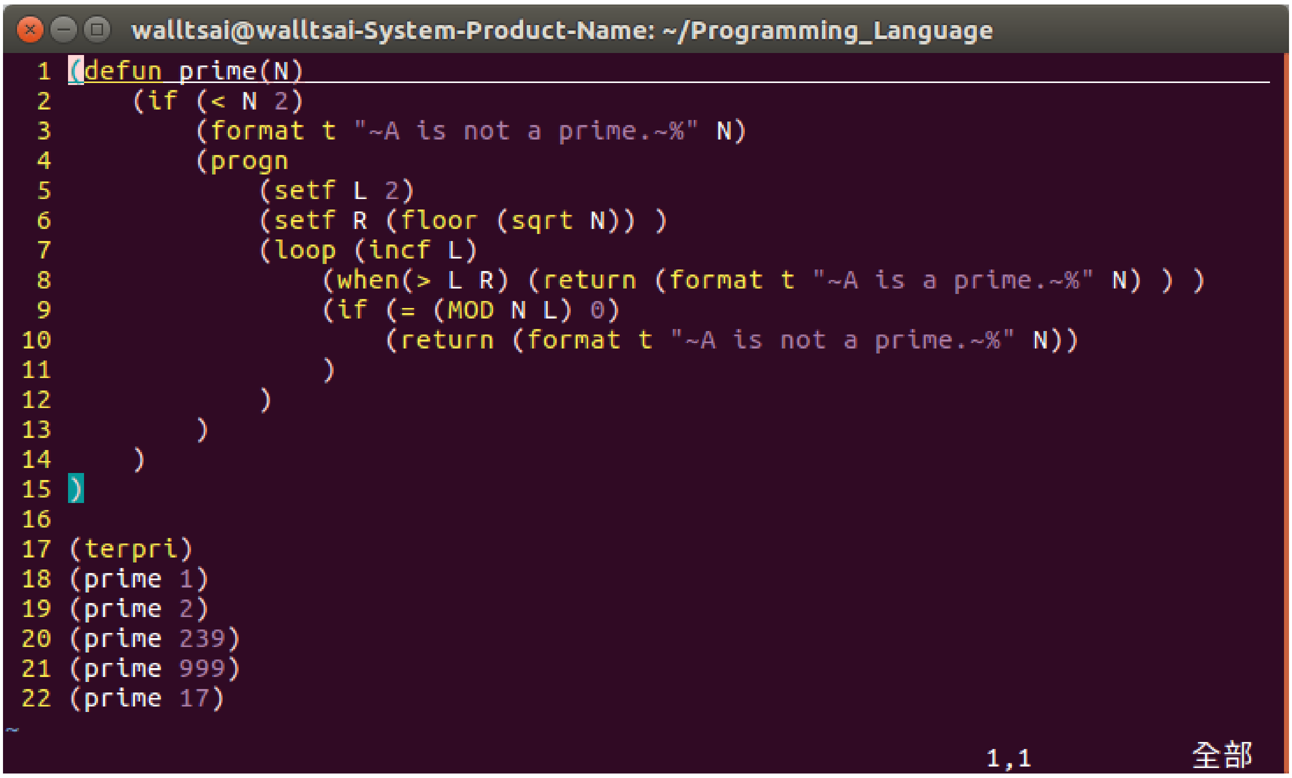
# HW1 LISP Report

執行環境：Ubuntu 16.04

## Problem 1.1


walltsai@walltsai-System-Product-Name: ~/Programming_Language
1,1
全部

```
(defun prime(N)
    (if (< N 2)
        (format t "~A is not a prime.~%" N)
        (progn
            (setf L 2)
            (setf R (floor (sqrt N)) )
            (loop (incf L)
                (when(> L R) (return (format t "~A is a prime.~%" N) ) )
                (if (= (MOD N L) 0)
                    (return (format t "~A is not a prime.~%" N))
                )
            )
        )
    )
)

(terpri)
(prime 1)
(prime 2)
(prime 239)
(prime 999)
(prime 17)
```

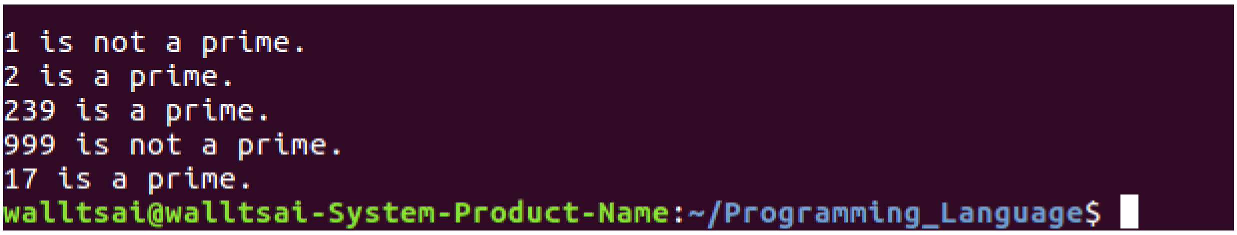
執行方式：$ sbcl --script prime.lsp

說明：

N 為要測試的數字

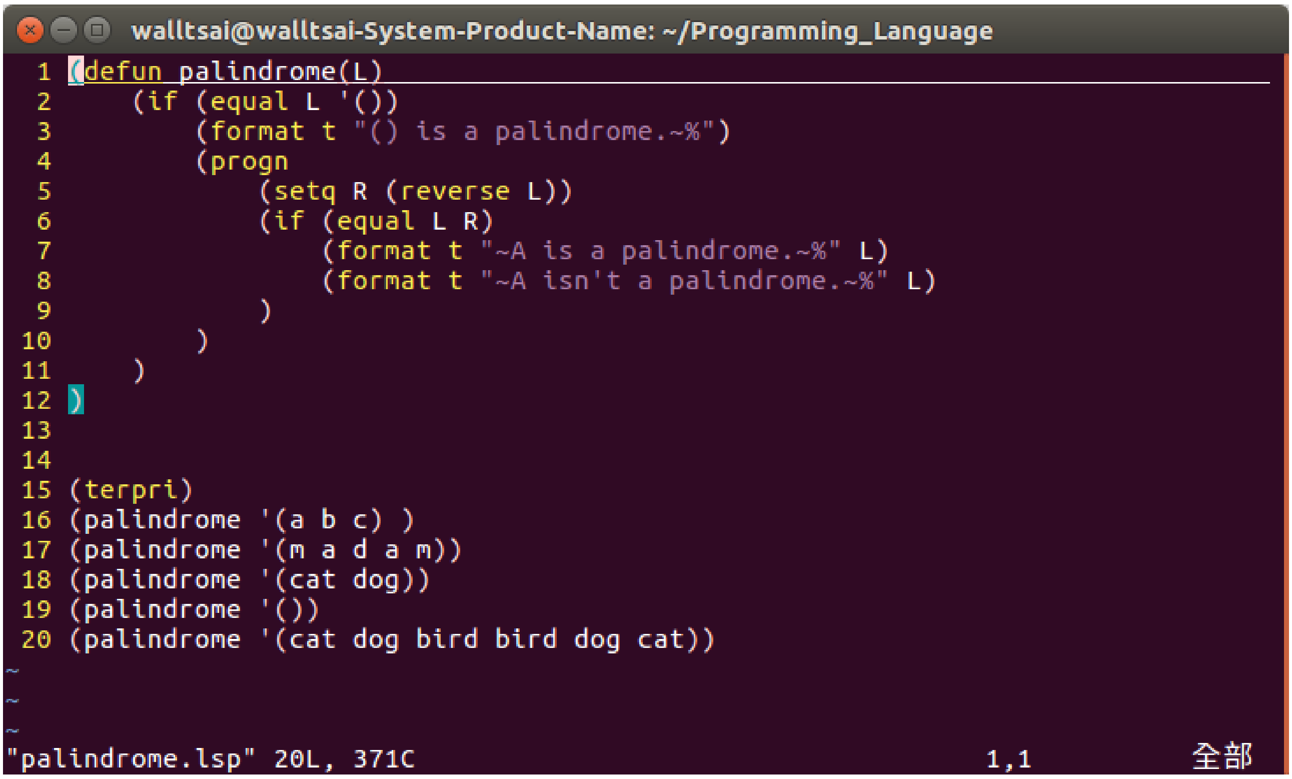
```
IF (N<2 (0 or 1)) THEN print (N is not prime.)
ELSE
    L = 2; R = floor(sqrt N)
    test every number from L to R
    if(N mod L == 0) THEN print (N is not prime.)
    if no number THEN print (N is a prime.)
```

Result:



```
1 is not a prime.
2 is a prime.
239 is a prime.
999 is not a prime.
17 is a prime.
walltsai@walltsai-System-Product-Name:~/Programming_Language$
```

## Problem 1.2

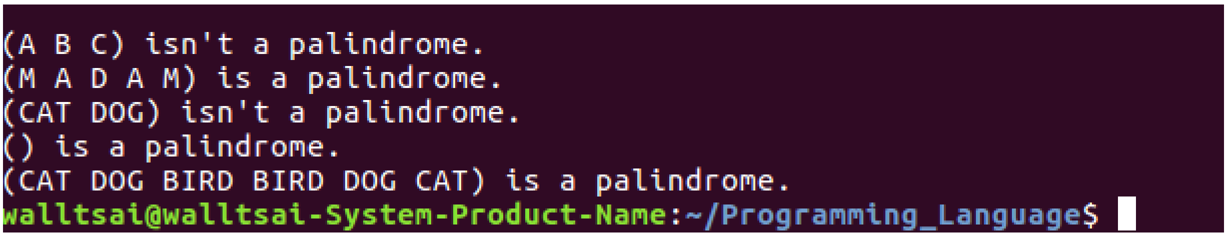

walltsai@walltsai-System-Product-Name: ~/Programming_Language

```
(defun palindrome(L)
    (if (equal L '())
        (format t "() is a palindrome.~%")
        (progn
            (setq R (reverse L))
            (if (equal L R)
                (format t "~A is a palindrome.~%" L)
                (format t "~A isn't a palindrome.~%" L)
            )
        )
    )
)


(terpri)
(palindrome '(a b c) )
(palindrome '(m a d a m))
(palindrome '(cat dog))
(palindrome '())
(palindrome '(cat dog bird bird dog cat))
~
~
~
"palindrome.lsp" 20L, 371C                                   1,1          全部
```

執行方式：$ sbcl --script palindrome.lsp
說明：
L 為要測試的 LIST

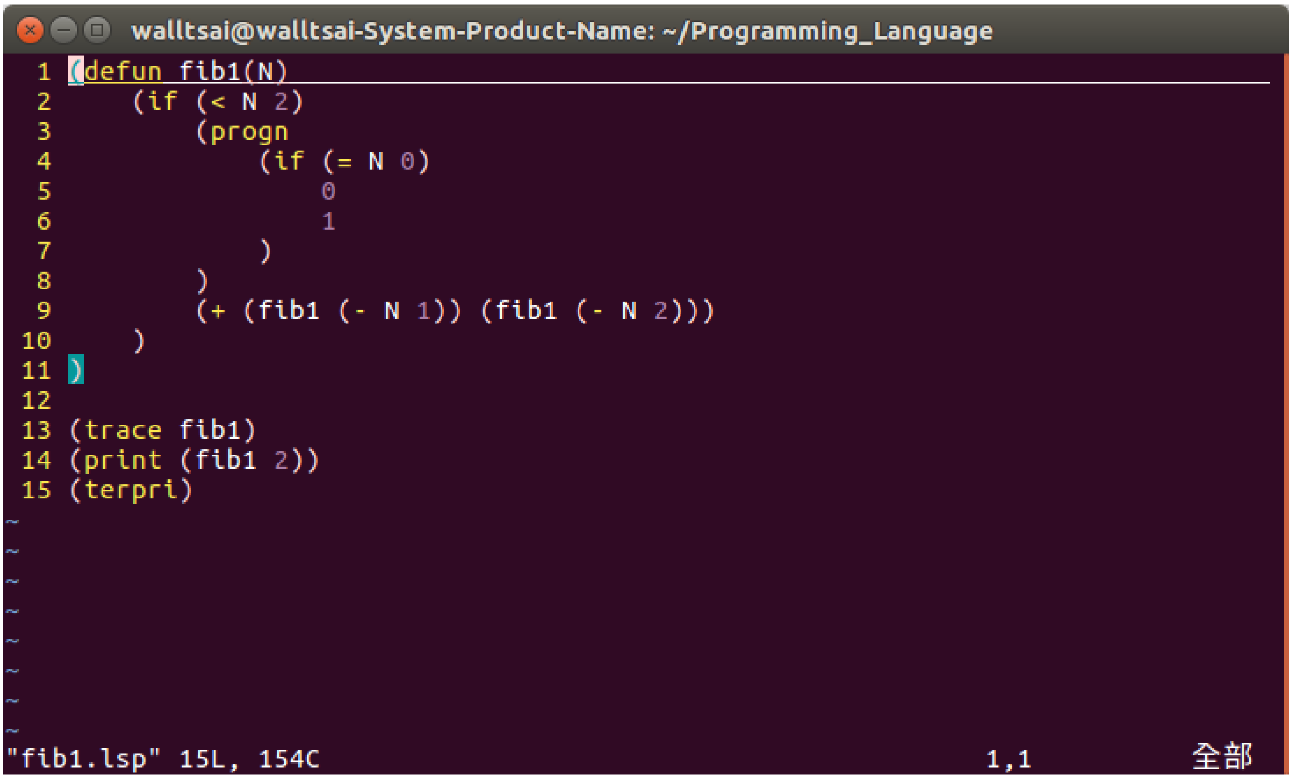
```
IF (L is empty LIST) THEN print (() is a palindrome.)
ELSE
    R = reverse(L)
    if(L == R) THEN print (() is a palindrome.)
    ELSE
        print (() isn’t a palindrome.)
```

Result:



```
(A B C) isn't a palindrome.
(M A D A M) is a palindrome.
(CAT DOG) isn't a palindrome.
() is a palindrome.
(CAT DOG BIRD BIRD DOG CAT) is a palindrome.
walltsai@walltsai-System-Product-Name:~/Programming_Language$
```

Problem 1.3



walltsai@walltsai-System-Product-Name: ~/Programming_Language

```
(defun fib1(N)
    (if (< N 2)
        (progn
            (if (= N 0)
                0
                1
            )
        )
        (+ (fib1 (- N 1)) (fib1 (- N 2)))
    )
)

(trace fib1)
(print (fib1 2))
(terpri)
~
~
~
~
~
~
~
~
"fib1.lsp" 15L, 154C                                         1,1          全部
```

fib1:

執行方式：$ sbcl --script fib1.lsp

說明：

N 代表 fibonacci 數列的第 N 項

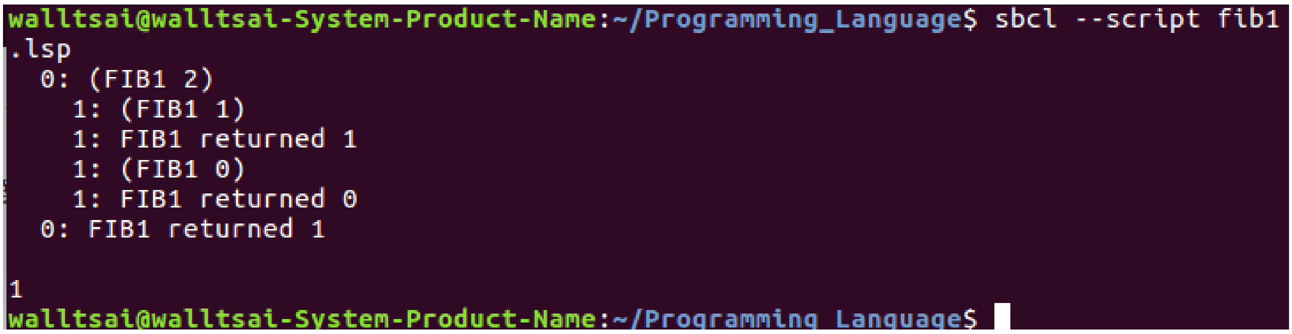
IF N==0 THEN return 0

IF N==1 THEN return 1

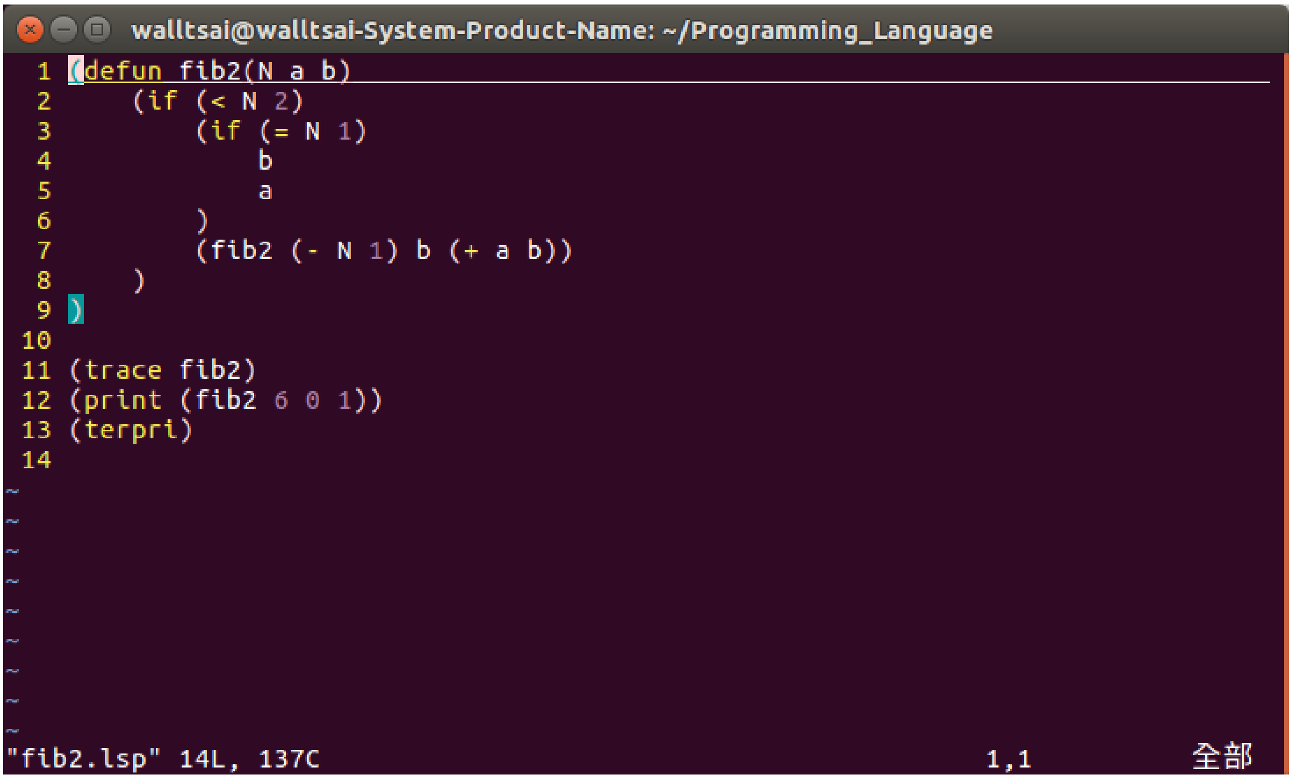
ELSE

return ( fib1(N-1) + fib1(N-2) )

Result:



```
walltsai@walltsai-System-Product-Name:~/Programming_Language$ sbcl --script fib1
.lsp
  0: (FIB1 2)
    1: (FIB1 1)
    1: FIB1 returned 1
    1: (FIB1 0)
    1: FIB1 returned 0
  0: FIB1 returned 1

1
walltsai@walltsai-System-Product-Name:~/Programming_Language$
```


walltsai@walltsai-System-Product-Name: ~/Programming_Language

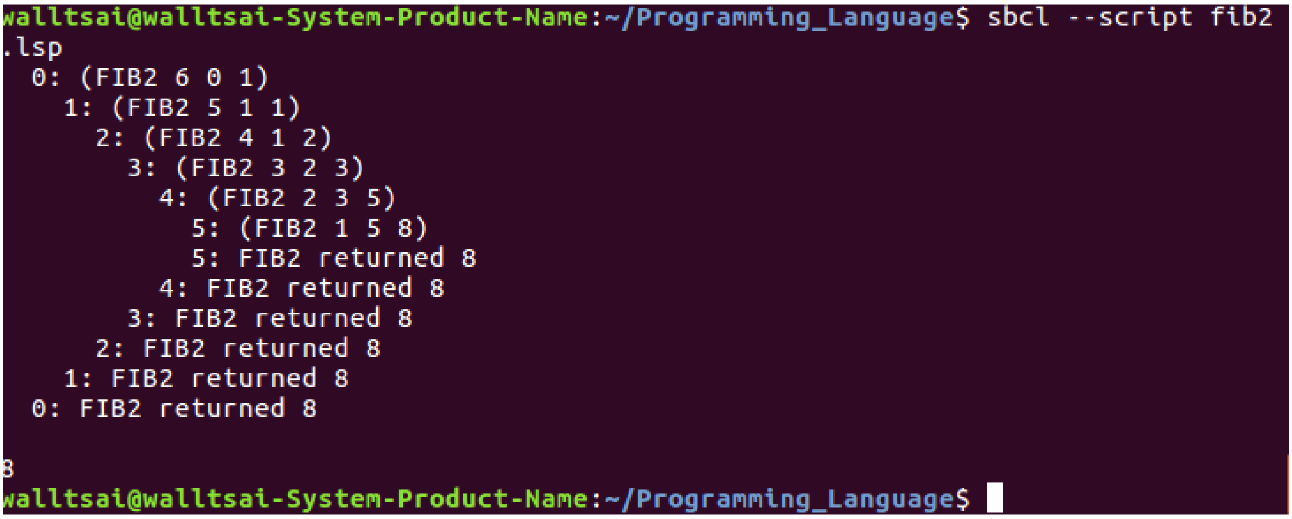
```
(defun fib2(N a b)
    (if (< N 2)
        (if (= N 1)
            b
            a
        )
        (fib2 (- N 1) b (+ a b))
    )
)

(trace fib2)
(print (fib2 6 0 1))
(terpri)

"fib2.lsp" 14L, 137C                                        1,1           全部
```

fib2:
執行方式：$ sbcl --script fib2.lsp
說明：
N 代表 fibonacci 數列的第 N 項，a 起始為 fibonacci 數列的第 0 項
b 起始為 fibonacci 數列的第 1 項

```
IF N=＝0 THEN return a
IF N==1 THEN return b
ELSE
    return fib2(N-1, b, a+b)
```

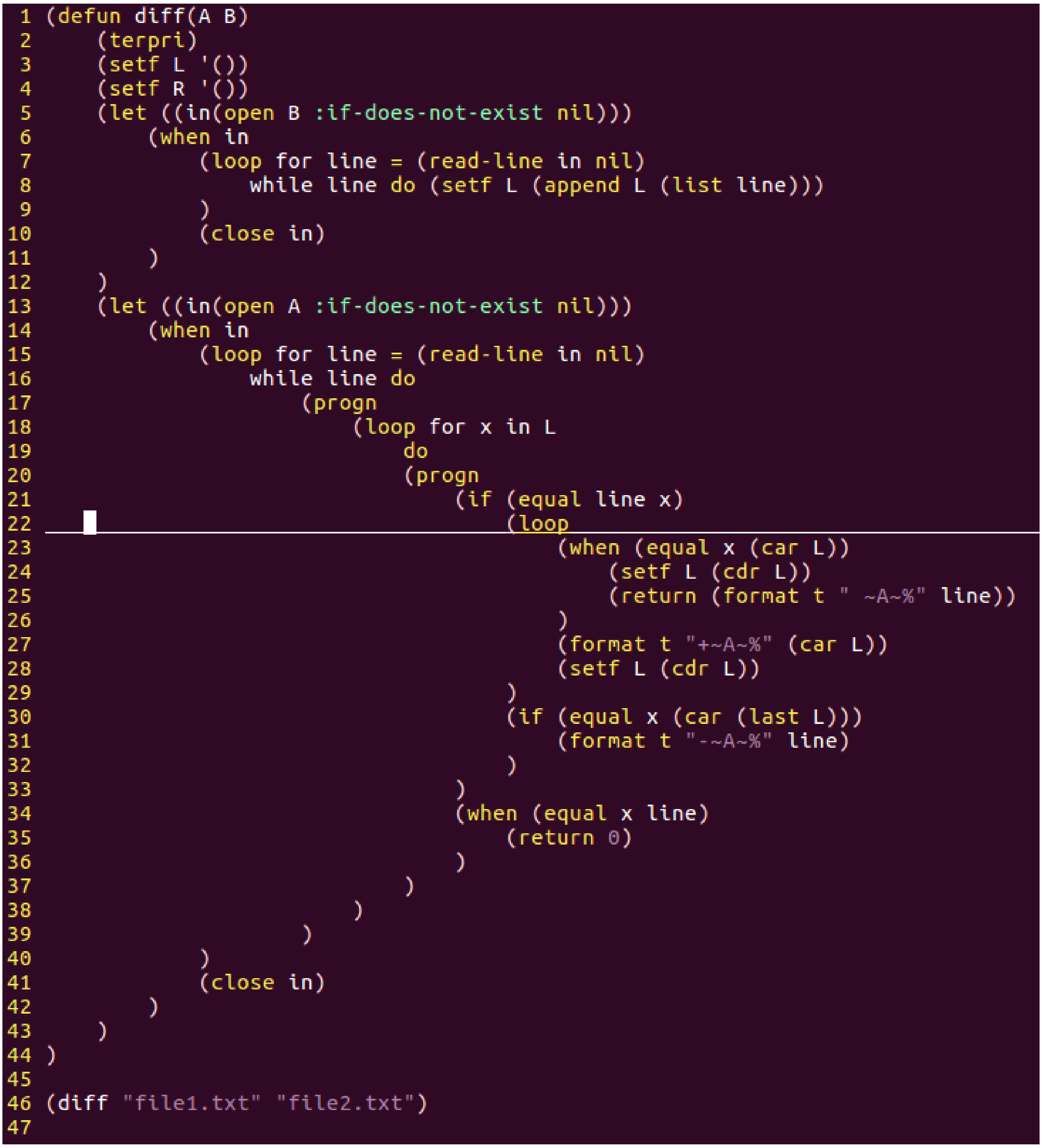
每遞迴一次，a 和 b 都會往下一項移動，N 則減一，所以 N==6 的話，遞迴到第 5 次時，就會是(FIB2　1　5　8)，這時 b 就會是第 6 項。

Result:



```
walltsai@walltsai-System-Product-Name:~/Programming_Language$ sbcl --script fib2
.lsp
  0: (FIB2 6 0 1)
    1: (FIB2 5 1 1)
      2: (FIB2 4 1 2)
        3: (FIB2 3 2 3)
          4: (FIB2 2 3 5)
            5: (FIB2 1 5 8)
            5: FIB2 returned 8
          4: FIB2 returned 8
        3: FIB2 returned 8
      2: FIB2 returned 8
    1: FIB2 returned 8
  0: FIB2 returned 8

8
walltsai@walltsai-System-Product-Name:~/Programming_Language$ 
```

## Problem 3



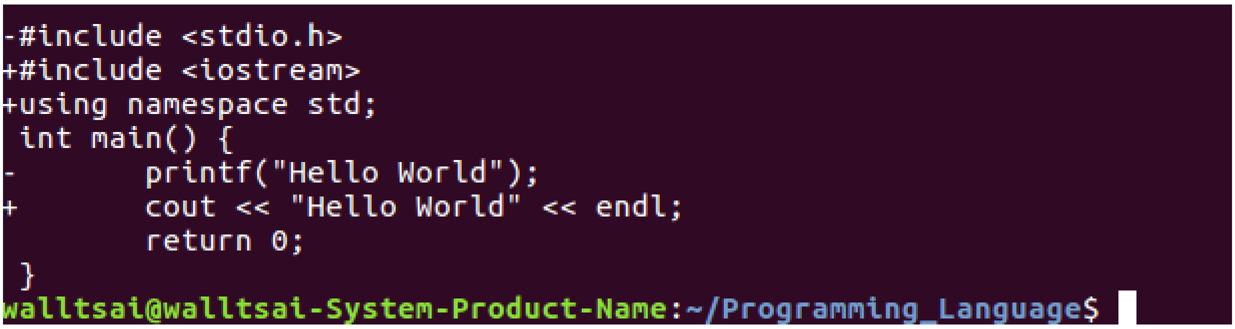
```lisp
(defun diff(A B)
    (terpri)
    (setf L '())
    (setf R '())
    (let ((in(open B :if-does-not-exist nil)))
        (when in
            (loop for line = (read-line in nil)
                while line do (setf L (append L (list line)))
            )
            (close in)
        )
    )
    (let ((in(open A :if-does-not-exist nil)))
        (when in
            (loop for line = (read-line in nil)
                while line do
                    (progn
                        (loop for x in L
                            do
                            (progn
                                (if (equal line x)
                                    (loop
                                        (when (equal x (car L))
                                            (setf L (cdr L))
                                            (return (format t " ~A~%" line))
                                        )
                                        (format t "+~A~%" (car L))
                                        (setf L (cdr L))
                                    )
                                    (if (equal x (car (last L)))
                                        (format t "-~A~%" line)
                                    )
                                )
                                (when (equal x line)
                                    (return 0)
                                )
                            )
                        )
                    )
            )
            (close in)
        )
    )
)

(diff "file1.txt" "file2.txt")
```

執行方式：$ sbcl --script diff.lsp
說明：
A 為第一個檔案 B 為第二個檔案

先將 B 的內容每一行存成一個列表，
再將 A 中的每一行，
和列表中每一個元素比較，
若沒有相同的，就print( “-～A” A的一行)，
若有相同的，

就把那行之前列表中的每個元素print(“+～A”元素)，
再print(“ ～A”相同的內容)，
再將列表中剩下的元素，存成一個新的列表，用來比較，
直到A沒有為止。

Result:



```
-#include <stdio.h>
+#include <iostream>
+using namespace std;
 int main() {
-       printf("Hello World");
+       cout << "Hello World" << endl;
        return 0;
 }
walltsai@walltsai-System-Product-Name:~/Programming_Language$
```