題目1-1：

執行環境：mac terminal (using vscode)

執行步驟：sbcl --script problem1-1.lsp

程式碼說明：

利用for迴圈從i=2到(i\*i >= input\_number)去取input\_number / i的餘數，若餘數等於0，則代表他不是質數而跳出。

(defun prime (input\_number) ; 定義input\_number

(let ((i 2)) ;令i=2

(loop

(cond ((> (\* i i) input\_number) ; if (i\*i >= input\_number)

(format t "True~%")

(return)))

(cond ((= (mod input\_number i) 0) ; 取input\_number / i的餘數，若餘數等於0

(format t "False~%")

(return)

))

(setq i (+ i 1))

)

)

)

題目1-2：

執行環境：mac terminal (using vscode)

執行步驟：sbcl --script problem1-2.lsp

程式碼說明：

去reverse input\_list，比較原list跟反過來的list是否相同即可。

(defun palindrome (input\_list)

(let ((tmpList (reverse input\_list))) ; reverse input\_list到tmplist

(if (equal tmpList input\_list) ; 如果相等就輸出true，用equal去比較數值而已

(format t "True~%")

(format t "False~%")

)

)

)

題目1-3：

執行環境：mac terminal (using vscode)

執行步驟：sbcl --script problem1-3.lsp

程式碼說明：

第一個是普通的迭代，第二個有回傳一個數值回去，我覺得有點像while迴圈去做i—1的條件而已，只是用迭代去實現。

(defun fib1 (n)

(if (<= n 2) ;如果n<=2回傳1，否則回傳fib1(n-1)+fib1(n-2)

1

(+ (fib1 (- n 1)) (fib1 (- n 2)))

)

)

(defun fib2 (n)

(defun tail-fib (n a b) ; 因為要在一個defun內做recursive，因此只能用labels

(if (= n 0) ; n是要找的第幾個數，a是a1，b是a2，最後的結果會存在a內

a

(tail-fib (- n 1) b (+ a b))

)

)

(tail-fib n 0 1);第一次迭代

)

(trace fib1)

(trace fib2)

題目2：

執行環境：mac terminal (using vscode)

執行步驟：sbcl --script mergesort.lsp

程式碼說明：

第一段就是單純的mergesort，利用資結學過的recursive版的mergesort，先寫出rmergeSort，再寫出listMerge即可實現mergesort。

第二段照著作業需求去應用mergesort。

(defun merge\_sort(lst)

(defun merge\_(f s)

(cond

((= (list-length f) 0) s)

((= (list-length s) 0) f)

((< (car f) (car s)) (append (list (car f)) (merge\_ (cdr f) s)))

((> (car f) (car s)) (append (list (car s)) (merge\_ f (cdr s))))

((= (car f) (car s)) (append (list (car f) (car s)) (merge\_ (cdr f) (cdr s))))

)

)

(let ((len (list-length lst)))

(cond

((= len 1) lst)

(t

(merge\_ (merge\_sort (subseq lst 0 (ceiling (/ len 2))))

(merge\_sort (subseq lst (ceiling (/ len 2))))

)

)

)

)

)

(let ((x 0) (tmplist ()))

(setq x (read)) ; 輸入要輸入幾個

(dotimes (i x) ; 要做幾次

(let ((tmp 0)) ; 把輸入得值存在list內

(setq tmp (read))

(push tmp tmplist)

)

)

(setq tmplist (reverse tmplist));因為用push所以要reverse回來

(dolist (n (merge\_sort tmplist));輸出mergesort跑完得list出來

(format t "~d " n))

(format t "~%")

)

題目3：

執行環境：mac terminal (using vscode)

執行步驟：sbcl --script diff.lsp

程式碼說明：

程式碼主要分成三部分，第一部分用於取出file內容，並逐行以一維list形式存在getIn1以及getIn2中。第二部分是一個函數（get\_list\_item），第一個參數=1指定要存取getIn1，第二個參數是要取出指定list中的第幾個。第三部分是主程式的部分，假設hello-world.c是file1.txt，hello-world.cpp是file2.txt，則中心思想是先用file1的第一行去比較file2的所有行，如果都沒有相同的，則輸出”- <此行內容>”，並取用file1下一行再去比，如果在file2第5行找到相同的行，則輸出1~4行的”+ <此行內容>”，而輸出第5行的”<此行內容>”，並下一次比較file2的行數更新從第6行開始，以此類推，則能搜尋完全部，但最後一行”}”要補上。

顏色輸出：(format t "~c[31m <想顯示的內容> [0m~%" #\ESC <變數>#\ESC)，而引號其中的31跟0，分別代表顏色顯示紅色，背景不變，若要顯示綠色則將31改成32。

;;;;;;;; Get All Needed Variable ;;;;;;;;

(defvar in1 (open "./res/file1.txt" :if-does-not-exist nil)) ;將file1內容存進去in1

(defvar countIn1 0) ;用於計算file1行數

(defvar getIn1 ()) ;用於將內容以一維list形式存在getIn1

(loop for line = (read-line in1 nil) ;迴圈逐行取出in1並暫存在line中

while line do (setq getIn1 (push line getIn1)) ;將line存到getIn1中

(setq countIn1 (+ countIn1 1)) ;順便使countIn1++以計算行數

)

(setq getIn1 (reverse getIn1)) ;由於是用push，因此結果要用reverse轉回來

;以下都跟file1相同

(defvar in2 (open "./res/file2.txt" :if-does-not-exist nil))

(defvar countIn2 0)

(defvar getIn2 ())

(loop for line = (read-line in2 nil)

while line do (setq getIn2 (push line getIn2))

(setq countIn2 (+ countIn2 1))

)

(setq getIn2 (reverse getIn2))

(close in1) ;關閉open指標

(close in2) ;關閉open指標

;;;;;;;; Utils ;;;;;;;;

(defvar return\_get\_list\_item "") ;由於不知道怎麼用return，因此設一個變數當作是return value

(defun get\_list\_item(list\_number number) ;number代表要取出list中的第幾個element

(let ((tmplist1 ())) ;宣告一個local 變數來做操作

(if (equal 1 list\_number) ; list\_number=1代表要取用getIn1

(setq tmplist1 getIn1)

(setq tmplist1 getIn2)

)

(setq return\_get\_list\_item (car tmplist1)) ;取出第一個值並存在return value中

(setq number (- number 1)) ;number數量-1

(if (> number 0) ;如果number一開始>1，就會往下做

(loop

(setq tmplist1 (cdr tmplist1)) ;取出剩下的放進tmplist1

(setq return\_get\_list\_item (car tmplist1));取出第一個值放return value中

(setq number (- number 1)) ;number數量-1

(when (equal number 0) (return)) ;一直重複直到number==0，就可取到要求位置的值

)

)

)

)

;;;;;;;; Main Function ;;;;;;;;

(defvar file2Line 1) ;紀錄file2要比較的位置的頭

(let ((i 1) (j 1)) ;宣告一個i=1, j=1

(loop

(get\_list\_item 1 i) ;取出第一個list的第i個行數內容

(let ((return\_list1 return\_get\_list\_item)) ;將取出值存在return\_list1

(block inner\_loop

(loop

(get\_list\_item 2 j) ;取出第二個list的第j個行數內容

(cond ((equal return\_list1 return\_get\_list\_item) ;比較左右行數內容

(loop ;如果相同

(get\_list\_item 2 file2Line) ;取出上一個比較的位置內容

(format t "~c[32m+ ~a~c[0m~%" #\ESC return\_get\_list\_item #\ESC) ;輸出綠色成”+ <想顯示的內容>”

(setq file2Line (+ 1 file2Line)) ;一直輸出直到上一個比較的位置等於目前相等的位置

(cond ((equal file2Line j) ;一但相等

(get\_list\_item 2 file2Line) ;取出相等位置內容

(write-line return\_get\_list\_item) ;正常輸出

(setq file2Line (+ 1 file2Line)) ;使上一個比較的位置更新到下一個位置

(return-from inner\_loop 0)) ;從最大的loop 離開

)

)

)

)

(setq j (+ 1 j)) ;若沒有相同就會跳進來這裡繼續比較file2下一行

(when (equal j countIn2) (return)) ;若比較到文末沒有相等一樣跳出迴圈

)

)

(cond ((equal j countIn2) ;若比較到文末沒有相等

(format t "~c[31m- ~a~c[0m~%" #\ESC return\_list1 #\ESC) ;輸出紅色成”- <想顯示的內容>”

)

)

(setq j file2Line) ;使j等於上一個比較的位置

)

(setq i (+ 1 i)) ;選擇file1下一行繼續比較

(when (equal i countIn1) (return)) ;比較到file1文末則比較結束

)

)

(format t "}~%") ;補上最後一個括號