Task 1

(B)
```
CL-USER> x1
2
CL-USER> x2
4
CL-USER> y1
3
CL-USER> y2
5
CL-USER> ( defun distance-between-points (x1 y1 x2 y2)
        (setf xdifference (- x2 x1))
        (setf ydifference (- y2 y1))
        (setf distance (sqrt (+ (expt xdifference 2) (expt ydifference 2))))
        distance
         )
; in: DEFUN DISTANCE-BETWEEN-POINTS
;     (BLOCK DISTANCE-BETWEEN-POINTS
;       (SETF XDIFFERENCE (- X2 X1))
;       (SETF YDIFFERENCE (- Y2 Y1))
;       (SETF DISTANCE (SQRT (+ # #)))
;       DISTANCE)
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::DISTANCE

;     (SETF DISTANCE (SQRT (+ (EXPT XDIFFERENCE 2) (EXPT YDIFFERENCE 2))))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::DISTANCE

;     (EXPT XDIFFERENCE 2)
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::XDIFFERENCE

;     (SETF XDIFFERENCE (- X2 X1))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::XDIFFERENCE

;     (EXPT YDIFFERENCE 2)
;
```

```
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::YDIFFERENCE

;     (SETF YDIFFERENCE (- Y2 Y1))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::YDIFFERENCE
;
; compilation unit finished
;   Undefined variables:
;     DISTANCE XDIFFERENCE YDIFFERENCE
;   caught 6 WARNING conditions
DISTANCE-BETWEEN-POINTS
CL-USER> (distance-between-points x1 y1 x2 y2)
2.828427
CL-USER>
```

(C)
```
CL-USER> (setf alist-sol
            (car (cdr (cdr '(A B X C D))))
            )
; in: SETF ALIST-SOL
;     (SETF ALIST-SOL (CAR (CDR (CDR '#))))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::ALIST-SOL
;
; compilation unit finished
;   Undefined variable:
;     ALIST-SOL
;   caught 1 WARNING condition
X
CL-USER> (setf blist-sol
            (car (car (cdr (cdr '(A B (X) C D)))))
            )
; in: SETF BLIST-SOL
;     (SETF BLIST-SOL (CAR (CAR (CDR #))))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::BLIST-SOL
;
; compilation unit finished
;   Undefined variable:
;     BLIST-SOL
```

```
;   caught 1 WARNING condition
X
CL-USER> (setf clist-sol
            (car (cdr (car (cdr '(A (B X C D)))))))
            )
; in: SETF CLIST-SOL
;    (SETF CLIST-SOL (CAR (CDR (CAR #))))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::CLIST-SOL
;
; compilation unit finished
;   Undefined variable:
;     CLIST-SOL
;   caught 1 WARNING condition
X
CL-USER> alist-sol
X
CL-USER> blist-sol
X
CL-USER> clist-sol
X
CL-USER>
```

(D)
```
CL-USER> (setf colors '(red white blue green gold brown purple))
; in: SETF COLORS
;    (SETF COLORS '(RED WHITE BLUE GREEN GOLD BROWN PURPLE))
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::COLORS
;
; compilation unit finished
;   Undefined variable:
;     COLORS
;   caught 1 WARNING condition
(RED WHITE BLUE GREEN GOLD BROWN PURPLE)
CL-USER> (qoute colors)
; in: QOUTE COLORS
;    (QOUTE COLORS)
;
; caught WARNING:
;   undefined variable: COMMON-LISP-USER::COLORS
;
```

```
; caught STYLE-WARNING:
;   undefined function: COMMON-LISP-USER::QOUTE
;
; compilation unit finished
;   Undefined function:
;     QOUTE
;   Undefined variable:
;     COLORS
;   caught 1 WARNING condition
;   caught 1 STYLE-WARNING condition
; Evaluation aborted on #<UNDEFINED-FUNCTION QOUTE {1004889E03}>.
CL-USER> (quote colors)
COLORS
CL-USER> 'colors
COLORS
CL-USER> colors
(RED WHITE BLUE GREEN GOLD BROWN PURPLE)
CL-USER> (desicribe 'colors)
; in: DESICRIBE 'COLORS
;     (DESICRIBE 'COLORS)
;
; caught STYLE-WARNING:
;   undefined function: COMMON-LISP-USER::DESICRIBE
;
; compilation unit finished
;   Undefined function:
;     DESICRIBE
;   caught 1 STYLE-WARNING condition
; Evaluation aborted on #<UNDEFINED-FUNCTION DESICRIBE {1004A5B433}>.
CL-USER> (describe colors)
(RED WHITE BLUE GREEN GOLD BROWN PURPLE)
  [list]
; No value
CL-USER> (describe 'colors)
COMMON-LISP-USER::COLORS
  [symbol]

COLORS names an undefined variable:
  Value: (RED WHITE BLUE GREEN GOLD BROWN PURPLE)
; No value
CL-USER> (type-of 'colors)
SYMBOL
CL-USER> (type-of colors)
CONS
```

CL-USER> (typep colors 'cons)
T
CL-USER> (typep colors 'list)
T
CL-USER> (typep colors 'symblol)
; Evaluation aborted on #<SIMPLE-ERROR "unknown type specifier: ~S" {1005140FB3}>.
CL-USER> (typep colors 'symbol)
NIL
CL-USER> (typep 'colors 'cons)
NIL
CL-USER> (typep 'colors 'symbol)
T
CL-USER>

(E)
CL-USER> (setf clubs '((10 C) (J C) (Q C) (K C) (A C)))
((10 C) (J C) (Q C) (K C) (A C))

CL-USER> clubs
((10 C) (J C) (Q C) (K C) (A C))

CL-USER> (setf diamonds '((10 D) (J D) (Q D) (K D) (A D)))
((10 D) (J D) (Q D) (K D) (A D))

CL-USER> diamonds
((10 D) (J D) (Q D) (K D) (A D))

CL-USER> (setf hearts '((10 H) (J H) (Q H) (K H) (A H)))
((10 H) (J H) (Q H) (K H) (A H))

CL-USER> hearts
((10 H) (J H) (Q H) (K H) (A H))

CL-USER> (setf spades '((10 S) (J S) (Q S) (K S) (A S)))
((10 S) (J S) (Q S) (K S) (A S))

CL-USER> spades
((10 S) (J S) (Q S) (K S) (A S))

CL-USER> (setf deck (append clubs diamonds hearts spades))
((10 C) (J C) (Q C) (K C) (A C) (10 D) (J D) (Q D) (K D) (A D) (10 H) (J H)
 (Q H) (K H) (A H) (10 S) (J S) (Q S) (K S) (A S))

CL-USER> deck

```
((10 C) (J C) (Q C) (K C) (A C) (10 D) (J D) (Q D) (K D) (A D) (10 H) (J H)
 (Q H) (K H) (A H) (10 S) (J S) (Q S) (K S) (A S))

CL-USER> (setf partitioned-deck (list clubs diamonds hearts spades))
(((10 C) (J C) (Q C) (K C) (A C)) ((10 D) (J D) (Q D) (K D) (A D))
 ((10 H) (J H) (Q H) (K H) (A H)) ((10 S) (J S) (Q S) (K S) (A S)))

CL-USER> partitioned-deck
(((10 C) (J C) (Q C) (K C) (A C)) ((10 D) (J D) (Q D) (K D) (A D))
 ((10 H) (J H) (Q H) (K H) (A H)) ((10 S) (J S) (Q S) (K S) (A S)))
CL-USER>
```

Task 2

```lisp
( defun distance (x1 y1 x2 y2)
        (setf x-difference (- x2 x1))
        (setf y-difference (- y2 y1))
        (sqrt (+ (expt x-difference 2) (expt y-difference 2)))
)
```

```
CL-USER> (load "perimeter_of_triangle.lisp")
T
CL-USER> (distance 5 4 3 2)
2.828427
CL-USER> (distance 9 8 7 6)
2.828427
CL-USER> (distance 100 140 450 502)
503.53152
CL-USER> (distance 64 52 62 54)
2.828427
CL-USER> (distance 4544 48452 4845 484)
47968.945
CL-USER>
```

Task 3

```lisp
(defun perimeter (x1 y1 x2 y2 x3 y3)
      (+
              (distance x1 y1 x2 y2)
              (distance x2 y2 x3 y3)
              (distance x3 y3 x1 y1)
```

```
        )
)
```

CL-USER> (load "perimeter_of_triangle.lisp")
T
CL-USER> (distance 5 4 3 2)
2.828427
CL-USER> (distance 9 8 7 6)
2.828427
CL-USER> (distance 100 140 450 502)
503.53152
CL-USER> (distance 64 52 62 54)
2.828427
CL-USER> (distance 4544 48452 4845 484)
47968.945
CL-USER> (load "perimeter_of_triangle.lisp")
T
CL-USER> (perimeter -3 6 -3 2 3 2)
17.211102
CL-USER> (perimeter 1 2 3 -4 -4 5)
23.557262
CL-USER> (perimeter -5 -2 -2 -2 -5 2)
12.0
CL-USER> (perimeter 130 402 346 893 539 392)
1482.4225
CL-USER>

Task 4

```lisp
( defun calculator-LR (&aux input operator operands prefix result)
       (format t "Expression? ")
       (setf input (list (read) (read) (read) (read) (read)))
       (setf first-operator (second input))
       (setf second-operator (nth 3 input))
       (setf operands (list (nth 0 input) (nth 2 input)))
       (setf third-operand (last input))
       (setf ex1 (apply first-operator operands))
       (setf ex2-operands (cons ex1 third-operand))
       (setf ex2 (apply second-operator ex2-operands))
       (format t "Result: ~A~%" ex2)
       (calculator-LR)
       )
```

CL-USER> (load "two_calculators.lisp")
T
CL-USER> (calculator-LR)
Expression? 5 * 5 + 2
Result: 27
Expression? 6 + 6 * 10
Result: 120
Expression? 12.3 + 4.5 * 9.6
Result: 161.28
Expression? 12.3 / 4.5 - 9.6
Result: -6.866667

Task 5

```
( defun calculator-RL (&aux input operator operands prefix result)
        (format t "Expression? ")
        (setf input (list (read) (read) (read) (read) (read)))
        (setf second-operator (nth 3 input))
        (setf second-operands (list (nth 2 input) (nth 4 input)))
        (setf ex2 (funcall second-operator (nth 2 input) (nth 4 input)))
        (setf first-operand (first input))
        (setf first-operator (second input))
        (setf ex1 (funcall first-operator first-operand ex2))
        (format t "Result: ~A~%" ex1)
        (calculator-RL)
        )

CL-USER> (calculator-RL)
Expression? 5 * 5 + 5
Result: 50
Expression? 12.3 + 4.5 * 9.6
Result: 55.5
Expression? 12.3 / 4.5 - 9.6
Result: -2.4117646
```