KONG Exam 1 (Version 3)

1.[1pt] Which one of the following is a correct way to write **''(A)** without using the ' character? Circle the only correct answer:

(a) (QUOTE (QUOTE A)) (b) (QUOTE QUOTE A) (c) (QUOTE (QUOTE (A)))
(d) (QUOTE QUOTE (A)) (e) (QUOTE ((QUOTE A)))

2.[1pt] What is the value of the lisp expression **(cons '(+ 3 4) '(+ 3 4))**? Circle the only correct answer:

(a) ((+ 3 4) + 3 4) (b) (+ 3 4 + 3 4) (c) (7 7)
(d) (7 + 3 4) (e) ((+ 3 4) (+ 3 4))

3.[1pt] What ist he value of the Lisp expression **(append '(+ 3 4) '(+ 3 4))**? Circle the only correct answer:

(a) ((+ 3 4) + 3 4) (b) (+ 3 4 + 3 4) (c) (7 7)
(d) (7 + 3 4) (e) ((+ 3 4) (+ 3 4))

4.[1pt] What is the value of the Lisp expression **(list '(+ 3 4) '(+ 3 4))**? Circle the only correct answer:

(a) ((+ 3 4) + 3 4) (b) (+ 3 4 + 3 4) (c) (7 7)
(d) (7 + 3 4) (e) ((+ 3 4) (+ 3 4))

5. Suppose the Lisp variable E has been given a value as follows:

```
(setf E '((2 Q) 9 19 29 39 49 59 69) (90 91 92 93 94 95 96 97)))
```

(i).[1pt] Write a Lisp expression that does not involve any numbers, but which evaluates to the following list: **(9 19 29 39 49 59 69)**

(ii).[1pt] Write a Lisp expression that does not involve any numbers, but which evaluates to the following list: **(92 93 94 95 96 97)**

(iii).[1pt] Write a Lisp expression that does not involve any numbers, but which evaluates to the number **2**.

(iv).[1pt] Suppose the Lisp variables A, B, and C have been given the values as follows:

```
(setf A '(9 19 29 39 49 59 69)
(set B '(92 93 94 95 96 97)
(set C 2)
```

Write a Lisp function that does not involve any numbers, but which evaluates to the following list: **((2 9 19 29 39 49 59 69) (92 93 94 95 96 97))**

6.[2pt] A point **(x, y)** in the plane can be represented as a 2-element list (x y). Complete the following definition of a Lisp function DST that takes two such lists p1 and p2 as arguments and returns the distance between the corresponding points. (Recall that the distance between two points (x1, y1) and (x2, y2) is given by sqrt of x1-x2 squared plus y1-y2 squared, and that the built in function **SQRT** returns the square root of its argument.

Example: **(dst '(4.0 6.0) '(1.0 2.0) => 5.0**

```
    (defun dst (p1 p2)

        (let ((xDiff (- __________  __________ ))

            (yDiff (- __________ __________ )))

            (sqrt (+ (* xDiff xDiff) (* yDiff yDiff)))))
```

7. Suppose the expressions (A)-(D) below are evaluated by Lisp immediately after evaluation of the following SETF expression: **(SETF L '(1 3 5 7))**. Write down the value of each of (A)-(D). [Be careful to write parentheses where they should be and nowhere else! You will receive no credit if the right answer is (z) and you write z.

(A) (MAPCAR #' (LAMBDA (I) (= I 5)) L) **ANSWER:**__________________[0.5pt]

(B) (MAPCAR #' (LAMBDA (X Y) (= X Y)) L L) **ANSWER:**__________________[0.5pt]

(C) (REMOVE-IF #' (LAMBDA (I) (= I 5)) L) **ANSWER:**__________________[0.5pt]

(D) (APPLY #'+ L) **ANSWER:**__________________[0.5pt]

8.[3pt] Complete the following definition of a Lisp function MONTH->INTEGER that takes as a argument a symbol that should be the name of a month, and that returns the number of the month. Examples: (MONTH->INTEGER 'FEB) => 2. If the argument is not a symbol, or if it is a symbol that is not the name of a month, then your function must return the symbol 'ERR.

```
    (defun month->integer (m) ;;; complete the rest of the code
```

9.[2pt] Define a recursive function SET-REMOVE such that if s is a list of numbers and/or symbols in which no atom occurs more than once, and y is an atom in s, then (SET-REMOVE y s) returns a list that consists of all the elements of s except y; but if y is not in s then (SET-REMOVE y s) just returns the list s. Examples:

```
(set-remove 'a           NIL) =>       NIL
(set-remove 'a    (a e i o u)) => (e i o u)
(set-remove 'a      (1 2 a 3)) =>   (1 2 3)
(set-remove 'a    (b 1 2 a 3)) => (b 1 2 3)
```

```
(defun set-remove (y s)
   (cond
      ((endp s) nil)
      ((eql y (car s)) ______________ )
      (t ( ______________  ______________ (set-remove y ______________ )))))
```

10. (i)[2pt.] Fill in the gaps G1 and G2 below to complete the following definition of a Lisp function REMOVE-ADJ-DUPL such that if the argument passed to REMOVE-ADJ-DUPL is a list of atoms then REMOVE-ADJ-DUPL returns a list obtained from the argument by removing all but one member of each sequence of adjacent duplicates in the argument.

```
Examples:   A. (REMOVE-ADJ-DUPL                              NIL) =>             NIL
            B. (REMOVE-ADJ-DUPL                             '(K)) =>             (K)
            C. (REMOVE-ADJ-DUPL        '(P A A D D D D C C A B B)) =>   (P A D C A B)
            B. (REMOVE-ADJ-DUPL      '(P P A A D D D D C C A B B)) =>   (P A D C A B)
            B. (REMOVE-ADJ-DUPL      '(Q P A A D D D D C C A B B)) => (Q P A D C A B)
```

The expression you write in gaps G1 and G2 may contain X, but must not call REMOVE-ADJ-DUPL

```
(defun remove-adj-dupl (L)
   (if (endp (cdr L))
      L
      (let ((X (remove-adj-dupl (cdr))))
         (if (equal (car L) (cadr L))
            ______________                        ;;; GAP G1
            ______________  _______________ )))) ;;; GAP G2
```

(ii).[1pt] Write a definition of the function REMOVE-ADJ-DUPL without using LET or LET*.

11.[1pt.] Compute the value of the following expression (which is written in prefix notation):

+4 3 -2 *3 3 -2 4 2 5 7 -2 3 1

Here +4 denotes the addition operator of arity 4 (e.q., the value of +4 3 2 7 5 is 17), *3 denotes the multiplication operator of arity 3 (e.g., the value of *3 3 2 7 is 42), and -2 denotes the subtraction operator of arity 2 (e.q., the value of -2 7 3 is 4).
Circle the correct value:

The value is: (a) -19 (b) -3 (c) 25 (d) 29 (e) 31

12. Here us a prefix expression ~3 x &2 ^3 y #2 z 2 p 7 $2 w v

The operators ~3 and ^3 are 3-ary (i.e., the arity of each is 3), but all of the other operators (&2 #2 $2) are binary.

(i) Draw the abstract syntax tree (AST) of this expression.

(ii) Rewrite the expression in postfix notation:

ANSWER: ______________________________ [1.5pt]