CS 343 – Structure of Programming Languages
Winter 2011, 4/8/2011

Programming Assignment in LISP
Due Date: Friday, April 22, 2011

1. Define a recursive function called `gcd` that takes two positive integers and returns the greatest common divisor of these integers using the Euclid's algorithm. The `gcd` is defined as follows: `gcd(a, 0) = a, gcd(a, b) = gcd(b, a mod b).`

```
(gcd 42 56)       returns 14
(gcd 12 18)       returns 6
(gcd 3 5)         returns 1
(gcd 12 60)       returns 12
(gcd 12 90)       returns 6
```

2. Define a function called `is-palindrome` and determines if the list is a palindrome. A palindrome is a sequence of things that can be read the same in either direction For example,

```
(is-palindrome ‘(a b a))         returns T
(is-palindrome ‘(a b))           returns NIL
(is-palindrome ‘(a))             returns T
(is-palindrome ‘())              returns NIL
```

3. Define a function called `intlist` that takes a positive integer and returns a list of all integers between 1 and the value of the argument, in ascending order. For example,

```
(intlist 8)  returns   (1 2 3 4 5 6 7 8)
```

You should count down in this function, so that you can just insert each new member into the front of the result variable. You can use recursion or iteration in your solution.

4. Define a function called `analyze` that takes a list and returns a list consisting of symbols *atom* and *list*. You can use recursion or iteration in your solution. For example,

```
(analyze ‘(a b c))           returns (atom atom atom)
(analyze ‘(a b (c d) e f))   returns (atom atom list atom atom)
(analyze ‘(a))               returns (atom)
(analyze ‘((a)))             returns (list)
```

5. Define a recursive function called `only-atoms` that takes a list as its single argument and returns T if the list contains only atomic values. It should return NIL if the list contains any non-atomic values (e.g., sublists).

```
(only-atoms '(a b c))        returns T
(only-atoms '(a))            returns T
(only-atoms '(a (b) c))      returns NIL
(only-atoms '())             returns NIL
```

6. Write the same function in Question 5 using a looping (iterative) construct. You can use either LOOP, DO, DOTIMES, or DOLIST in your solution.

```
(only-atoms-iter '(a b c))       returns T
(only-atoms-iter '(a))           returns T
(only-atoms-iter '(a (b) c))     returns NIL
(only-atoms-iter '())            returns NIL
```

7. Define a function called `quad-roots` that takes three parameters *a*, *b*, and *c*, and returns a list containing the two roots of the quadratic equation `ax^2 + bx + c = 0`. For example,

```
(quad-roots 2 4 -30)   returns (3 -5)
(quad-roots 1 3 -4)    returns (-4 1)
(quad-roots 2 -4 -3)   returns (-058 2.58)
```

8. Define a function called `rotate-left` that takes a list and an integer as arguments and returns a new list with *n* elements rotated to the left. You can use recursion or iteration in your solution. For example,

```
(rotate-left ‘(a b c d e) 3)     returns (D E A B C)
(rotate-left '(a b c d e) 2)     returns (C D E A B)
(rotate-left '(a b c d e) 1)     returns (B C D E A)
(rotate-left '(a b c d e) 4)     returns (E A B C D)
(rotate-left '(a b c d e) 5)     returns (A B C D E)
(rotate-left '(a b c d e) 7)     returns (C D E A B)
```

9. Write a recursive function `evens` that returns every other element of a list, beginning with the second. That is, the function returns all the elements in even-numbered positions in the list. For example,

```
(evens ‘(a b c d e))        returns (B D)
(evens '(a))                returns NIL
(evens '(a b))              returns (B)
(evens '())                 returns NIL
(evens '(a b c d e f g))    returns (B D F)
```

10. Write the same function in Question 9 using a looping (iterative) construct. You can use either LOOP, DO, DOTIMES, or DOLIST in your solution. For example,

```
(evens-iter ‘(a b c d e))        returns (B D)
(evens-iter '(a))                returns NIL
(evens-iter '(a b))              returns (B)
(evens-iter '())                 returns NIL
(evens-iter '(a b c d e f g))    returns (B D F)
```

Deliverables

1. Define these functions in a text file named **Project6.lsp**
2. Upload the file **Project6.lsp** on Blackboard.
    - I will use the submission date/time on Blackboard as your official submission date/time.
    - It is your responsibility to make sure the submission on Blackboard went through successfully.
3. Late submission: if you submit it on Monday, April 25th, you will lose a total of 10 points for the delayed submission.