**ASSIGNMENT 5**

This assignment is about writing functional programs using Lisp language. All functions must be written in Lisp that runs on the LispWorks software. You can only use those built in Lisp functions discussed in class. The functions must be placed in a file named program.lsp. The program must be well documented. Test your functions thoroughly with all input cases.

1. Write a function f1 that counts the number of lists in a given list L. `Example: (f1 '(a (a b) c d (e))) returns 2`

2. Write a function f2 that decides whether a given list L has an atom inside. `Example: (f2 '((a b)(c d))) returns nil, (f2 '(a (b c))) returns t`

3. Write a function f3 that takes a list of integers L and returns a list containing only odd integers in L. `Example: (f3 '(23 4 7 18 22)) returns (23 7)`

4. Write a function f4 that returns the last element of a given list L. `Example: (f4 '(a (b c) (c d))) returns (c d)`

5. Write a function f5 that returns the reverse of a given list L. `Example: (f5 '(a (b c) (x) d)) returns (d (x) (b c) a)`

6. Write a function f6 that returns the first list of the list. `Example: (f6 '(c (a b) d (x y))) returns (a b)`

7. Write a function f7 that returns the list containing the lists of a given list L. `Example: (f7 '(a (b c) d x (y))) returns ((b c)(y))`

8. Write a function f8 that returns the product of all integers everywhere in a given list L. `Example: (f8 '(2 (5 4) 3 (2 (1 10)) 5)) returns 12000`

9. Write a function f9 that removes duplicates from a given list L. `Example: (f9 '(c (a b) c d (a b))) returns (c (a b) d)`

10. Write a function f10 that finds the intersection of two lists L and M. The intersection here means the common elements of the two lists. `Example: (f10 '(a (a b) c d) '(b (a b) d)) returns ((a b) d)`

11. Write a function f11 that decides whether a given positive integer N is a prime number or not. `Example: (f11 43) returns t, (f11 24) returns nil`

12. Write a function f12 that sorts a list L. List elements are integers. Insertion sort must be used. `Example: (f12 '(4 9 2 7 6)) returns (2 4 6 7 9)`