Homework 2

1) What is the programming structure for LISP?

2) How data types are categorized in LISP?

3) Mention what are the three components/elements needed by LISP to write a function?

4) Which of the following is the difference between an expression and a statement?
(a) An expression has a value, while a statement does not;
(b) An expression does not modify anything, while a statement does;
(c) A pure expression does not have side effects, while a statement does.

5) The advantages of functional programming without modifications are
(a) complex data structures can be shared without unexpected side effects
(b) computations can be parallelized without unexpected interactions among threads
(c) programs can be more efficiently translated to machine code
(d) Less memory is required for the computation

6) Which of the following is not a feature of the Lisp programming language?
(a) Functional programming using expressions
(b) Using functions as first-class objects
(c) Dynamic interpretation of nested lists as program representations
(d) A memory model based on singly linked lists
(e) Allows users to define their own types of data structures

7) Lisp treats functions as first-class objects because it
(a) allows function definitions to be nested
(b) allows functions to be returned as result of other functions
(c) allows anonymous functions to be used as expressions
d) allows functions to be passed as parameters to other functions
(e) allows forward function declarations
(f) treats Functions as a type of built-in values

8) Which of the following are higher order functions?
(a) a function that takes another function as parameter
(b) a function that returns another function as result
(c) a function that includes another function definition inside its body
(d) a function that does not modify any variables

9) Write a function Map that takes two parameters, a function f, and an arbitrary value y. The Map function then returns the result of invoking f to modify each element contained inside y.

10) Define a function substitute which takes three parameters, x, y, and z. It returns a new list which replaces all occurrences of x in y with z.

11) Define in Lisp a function Find which takes two parameters, x and y. It returns x if x appears in y, and returns an empty list otherwise.

12) Define a function count, which takes an arbitrary parameter x and returns the number of numeric values (i.e., numbers) contained in x. Test your code with the following cases.

```
> (count 'x)
0
> (count '(x 3))
1
> (count '((1 2) 3) )
3
```

13) What is the difference between typing the following items at the interpreter's top level? In each case, tell what will be displayed and how the interpreter determined the information displayed.

    a. A
    b. 'A
    c. #\A