**Q1.1**

As discussed in lecture 5, there is no real need for binding (define) in programming languages, it only improves readability and convenience. It is recommended but not required and therefore, if L11 is defined as L1 excluding "define", every program that can be written in L1 can be written in L11. You can replace every "define" by copying and substituting the value manually wherever it is needed.

**Q1.2**

As discussed in lecture 5, there is no real need for binding (define) in programming languages, it only improves readability and convenience. It is recommended but not required and therefore, if L21 is defined as L2 excluding "define", every program that can be written in L2 can be written in L21. You can replace every "define" by copying and substituting the value manually wherever it is needed.

Recursions can work without 'define' by using a complex work around.

**Q1.3**

There are programs that can be written in L2 but not in L22 such as:

L2:

(lambda (x y) (+ x y))

The value of the program is a closure with two arguments and cannot be created in any way in L22.

**Q1.4**

There are programs that can be written in L2 but not in L23 such as the well-known function "map":

(define map

(lambda (f lst)

(if (eq? lst '())

'()

(cons (f (car lst)) (map f (cdr lst)))

)

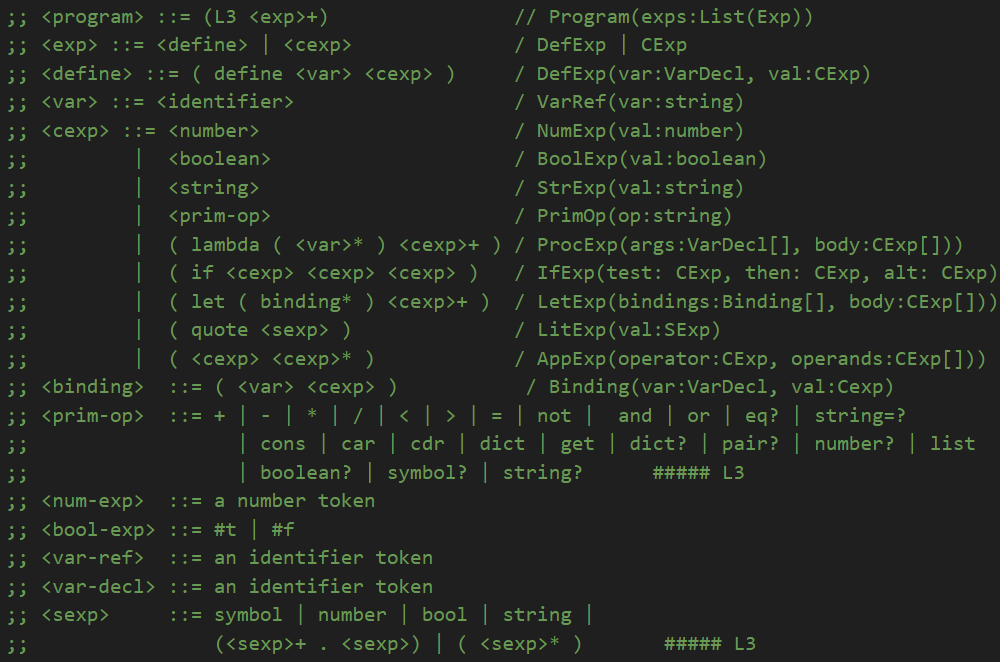
)

)

There is no single behavior for the function, it depends on the mapping function.  
With no high order functions each function would have only one defined behavior.

**Q2.1**

**b.**



Q2.2

