Part 1:

(+ 25 9 16)
50
(/ 24 4)
6
(+ (* 3 28) (- 2 2))
84
(define a 8)
Nothing
(define b (+ a 7))
Nothing
(+ a b (* a b))
143
(= a b)
False
(if (and (> b a) (< b (* a b))) b a)
15
(cond ((= a 9) 6) ((= b 3) (+ 6 7 a)) (else 25))
25
(+ 10 (if (> b a) b a))
25
(* (cond ((> a b) a) ((< a b) b) (else -1)) (+ a 15))
345

Part 2:

Part A:

(define (idx_getter l n) (cond ((null? l) l) ((= n 0) (car l)) (else (idx_getter (cdr l) (- n 1)))))

For the second part of the question if we want to create a sublist from ith element to the jth element. What we can do is we can first go the the ith element then we can create a pair with the ith +1 element and keep appending all the remaining elements until we reach the jth element. This will subsequently create a sublist from ith element to the jth element.

Part B:

(define (nth n) (cond ((= n 0) 1) ((> 0 n) 0) (else (+(*(nth (- n 1)) (nth (- n 1)))4))))

Part C:

```
(define (prime?-helper n x)
        (cond ((<= n 1) false)
        ((= n 2) true)
        ((= n x) true)
        ((= (modulo n x) 0) false)
        (else (prime?-helper n (+ x 1))) ) )
(define (prime? n) (prime?-helper n 2))
```