Problem 1.

Top-Down - reduce to base case.
Nonnegative Integer n is in S iff:
1. n = 0, or
2. (sqrt(n) - 1)^2 is in S

Bottom-Up - start from base case.
S is the smallest set contained in the set of nonnegative integers satisfying the following properties:
1. n = 0 and (n is in S), or
2. if (n is in S), then (n + 2*sqrt(n) + 1) is in S

Rules of Inference
Sequence of derivations (Induction Hypothesis + Inductive Step)
Base Case (always holds):
>> n is in S.
Rule of Derivation (by induction):
>> Assume k=m^2 is a square number. So, k is in S.
>> Then (sqrt(k)+1)^2 = (m+1)^2 is a square number and is in S.
Proof.
k = m^2 (ADD (2*m + 1) to both sides)
k + 2*m + 1 = m^2 + 2*m + 1 = (m+1)^2.

Problem 2.

```
(define (repeatN lst n)
  (if (null? lst) '()
      (let ((first (car lst)) (rest (cdr lst)))
        (append (make-list n first) (repeatN rest n)))))
```

Problem 3.

```
(define (sum-even lst) (reduce + 0 (filter even? lst)))
```

Problem 4.

```
(define (isIn? lst n)
  (or (eqv? (car lst) n)
      (and (not (null? (cdr lst)))
           (isIn? (cdr lst) n))))

(define (myProc pred lst1 lst2 op init)
  (fold-right op init (filter (lambda (e) (pred lst2 e)) lst1)))
```