Problem 1.

1. Proof By Induction

Base case $(n = 1)$

Left side: $\sum_{i=1}^{1} (2i-1)^2 = (1)^2 = 1$
Right side: $1 \cdot (2 \cdot 1 - 1) \cdot (2 \cdot 1 + 1) / 3 = (1 \cdot 1 \cdot 3)/3 = 1$
So $P(1)$ is true.

Assume $P(k)$ true for some $k \geq 1$:
$\sum_{i=1}^{k} (2i - 1)^2 = k(2k - 1)(2k + 1) / 3$

Show $P(k) \to P(k+1)$
$\sum_{i=1}^{k+1} (2i - 1)^2$
$= \left[\sum_{i=1}^{k} (2i - 1)^2\right] + (2(k+1) - 1)^2$

$k(2k - 1)(2k + 1)/3 + (2k + 1)^2$

Simplify $(2k + 1)^2 = 4k^2 + 4k + 1 = (12k^2 + 12k + 3)/3$

$[k(2k - 1)(2k + 1) + 12k^2 + 12k + 3] / 3$

Expand $k(2k - 1)(2k + 1) = k(4k^2 - 1) = 4k^3 - k$

The numerator becomes:
$4k^3 - k + 12k^2 + 12k + 3$
$= 4k^3 + 12k^2 + 11k + 3$

Now factor the numerator as $(k + 1)(2k + 1)(2k + 3)$:

$(k+1)(2k+1)(2k+3)$
$= (k+1)(4k^2 + 8k + 3)$
$= 4k^3 + 8k^2 + 3k + 4k^2 + 8k + 3$
$= 4k^3 + 12k^2 + 11k + 3$

$\sum_{i=1}^{k+1} (2i - 1)^2$
$= (k+1)(2k+1)(2k+3) / 3$
$2k+1 = 2(k+1) - 1$, and $2k+3 = 2(k+1) + 1$
$= (k+1)(2(k+1) - 1)(2(k+1) + 1) / 3$

$P(k+1)$ holds true.

Problem 2.

Formal Definition M = (Q, Sigma, delta, q0, F):

1. States (Q):
   Q = {qe, qo, q0s, q_trap}
   - qe: Even number of 1s seen
   - qo: Odd number of 1s seen.
   - q0s: 0s only segment
   - q_trap: Invalid string found.

2. Alphabet (Sigma):
   Sigma = {0, 1}

3. Start State (q0):
   q0 = qe

4. Accept States (F):
   F = {qe, q0s}

5. Transition Function (delta)
   - From State qe:
     1. - delta(qe, 0) = q0s
     2. - delta(qe, 1) = qo
   - From State qo:
     1. delta(qo, 0) = qo
     2. delta(qo, 1) = qe
   - From State q0s:
   - - delta(q0s, 0) = q0s
   - - delta(q0s, 1) = q_trap
   - From State q_trap:
     1. delta(q_trap, 0) = q_trap
     2. delta(q_trap, 1) = q_trap

Problem 3

States: Q = {q0, q1, q2, q3}
Start State: q0
Accept States (F): F = {q0, q3}

a. String: 0001
- Path: q0 ->(0) q0 ->(0) q0 ->(0) q0 ->(1) q1
- Final State: q1
- Result: Reject

b. String: 10011
- Path: q0 ->(1) q1 ->(0) q2 ->(0) q1 ->(1) q3 ->(1) q3
- Final State: q3
- Result: Accept

c. String: 1101
- Path: q0 ->(1) q1 ->(1) q3 ->(0) q2 ->(1) q0
- Final State: q0
- Result: Accept

d. String: 11100
- Path: q0 ->(1) q1 ->(1) q3 ->(1) q3 ->(0) q2 ->(0) q1
- Final State: q1
- Result: Reject

Problem 4

L1 = {w: 100 is not a substring of w}
L2 = {w: 2 <= |w| <= 4}

1. DFA M1 (for L1)

1. States (Q1):
   Q1 = {s, q1, q10, q100}

2. Start State (q0_1):
   q0_1 = s

3. Accept States (F1):
   F1 = {s, q1, q10}

4. Transitions (delta1):
   - delta1(s, 0) = s
   - delta1(s, 1) = q1
   - delta1(q1, 0) = q10
   - delta1(q1, 1) = q1
   - delta1(q10, 0) = q100 (100 seen)
   - delta1(q10, 1) = q1
   - delta1(q100, 0) = q100
   - delta1(q100, 1) = q100

2. DFA M2 (for L2)

1. States (Q2):
   Q2 = {t0, t1, t2, t3, t4}

2. Start State (q0_2):
   q0_2 = t0

3. Accept States (F2):
   F2 = {t2, t3, t4}

4. Transitions (delta2):

```
  - delta2(t0, 0/1) = t1
  - delta2(t1, 0/1) = t2
  - delta2(t2, 0/1) = t3
  - delta2(t3, 0/1) = t4
  - delta2(t4, 0/1) = t4
```

3. Union DFA M (for L = L1 U L2)

```
1. States (Q):
   Q = {(qA, qB) | qA in Q1, qB in Q2}

2. Start State (q0):
   q0 = (s, t0)

3. Accept States (F):
   (qA, qB) is accepted if qA is in F1 OR qB is in F2.
   F = {(qA, qB) | qA in {s, q1, q10} OR qB in {t2, t3, t4}}

4. Transitions (delta):
   delta((qA, qB), x) = (delta1(qA, x), delta2(qB, x))
```