Problem 1.

FORMAL DEFINITION OF DFA M
State set: QM
QM = {q0, q1, q2, q3}

Alphabet: Sigma
Sigma = {0, 1}

Start state: qM0
qM0 = q0

Accepting states: FM
FM = {q2}

Transition function delta_M:
delta_M(q0, 0) = q3
delta_M(q0, 1) = q1
delta_M(q1, 0) = q2
delta_M(q1, 1) = q1
delta_M(q2, 0) = q2
delta_M(q2, 1) = q2
delta_M(q3, 0) = q3
delta_M(q3, 1) = q3

FORMAL DEFINITION OF DFA N
State set: QN
QN = {r0, r1, r2}

Alphabet: Sigma
Sigma = {0, 1}

Start state: qN0
qN0 = r0

Accepting states: FN
FN = {r2}

Transition function delta_N:
delta_N(r0, 0) = r1
delta_N(r0, 1) = r0
delta_N(r1, 0) = r0
delta_N(r1, 1) = r2
delta_N(r2, 0) = r0

```
delta_N(r2, 1) = r0

FORMAL DEFINITION OF NFA P for (L1 union L2)*

P state set:
Q = QM union QN union {qs}
Q = {q0, q1, q2, q3, r0, r1, r2, qs}

P start state:
qs

P accepting states:
F = {qs} union FM union FN
F = {qs, q2, r2}

P transitions (delta):

Original transitions (for a in {0, 1}):
 For q in QM: delta(q, a) = {delta_M(q, a)}
 For q in QN: delta(q, a) = {delta_N(q, a)}
 delta(qs, a) = {} (empty set)

Epsilon transitions (for alpha = epsilon):
 Epsilon transitions from new start state:
  delta(qs, epsilon) = {q0, r0, qs}

 Loopback epsilon transitions from accepting states:
  For f in FM (f=q2): delta(q2, epsilon) = {qs}
  For f in FN (f=r2): delta(r2, epsilon) = {qs}

 For all other states q in Q:
  delta(q, epsilon) = {} (empty set)
```

Problem 2.

a. String: 1001

Check on NFA M
Path: q0 (1) --> q2 (0) --> q0 (0) --> q1 (1) --> q0
Result: Final state is {q0.
Answer: NFA M accepts 1001.

Check on NFA N
Path:q0 (1) --> {q2, q0} (0) --> {q1} (0) --> {q1, q3} (1) --> {q1, q0}
Result: Final set is {q1, q0}.
Answer: NFA N rejects 1001.

b. String: 11110

Check on NFA M
Path: q0 (1) --> q2 (1) --> q0 (1) --> q2 (1) --> q0 (0) --> q1
Result: Final state is {q1}.
Answer: NFA M rejects 11110.

Check on NFA N
Path:q0 (1) --> {q2, q0} (1) --> {q2, q0} (1) --> {q2, q0} (1) --> {q2, q0} (0) --> {q1}
Result: Final set is {q1}.
Answer: NFA N rejects 11110.

c. String: 1101

Check on NFA M
Path: q0 (1) --> q2 (1) --> q0 (0) --> q1 (1) --> q0
Result: Final state is {q0}.
Answer: NFA M accepts 1101.

Check on NFA N
Path:q0 (1) --> {q2, q0} (1) --> {q2, q0} (0) --> {q1} (1) --> {q1}
Result: Final set is {q1}.
Answer: NFA N rejects 1101.

d. String: 11100

Check on NFA M
Path: q0 (1) --> q2 (1) --> q0 (1) --> q2 (0) --> q0 (0) --> q1
Result: Final state is {q1}.
Answer: NFA M rejects 11100.

Check on NFA N
Path:q0 (1) --> {q2, q0} (1) --> {q2, q0} (1) --> {q2, q0} (0) --> {q1} (0) --> {q1, q3}
Result: Final set is {q1, q3}.
Answer: NFA N accepts 11100.

Problem 3.

a.
Members: 10, 110, 10100
Non-Members: 0, 1

b.
Members: 0, 11, 000
Non-Members: epsilon, 100

c.
Members: epsilon , 00, 001
Non-Members: 10, 110

d. Members: 001, 1011, 00101
Non-Members: 01, 100

e.
Members: 101000, 111010, 1011010
Non-Members: 10100, 001000

f.
Members: 0, 111, 010
Non-Members: epsilon, 00

Problem 4.

FORMAL DEFINITION OF DFA 3A

State set: {A, B, C, D, E, F}
Alphabet: {0, 1}
Start state: A
Accepting states: {F}

Transition function delta_A:

State A (Start):
A (0) --> B
A (1) --> C

State B:
B (0) --> B
B (1) --> B

State C:
C (0) --> D
C (1) --> C

State D:
D (0) --> D
D (1) --> E

State E:
E (0) --> F
E (1) --> E

State F:
F (0) --> F
F (1) --> B

Problem 5.

1. Initial GNFA Definition

The original DFA have states q0, q1, q2, q3.
The accepting states are {q1, q2}

Initial Transitions:
q_start (epsilon) --> q0
q0 (1) --> q1
q0 (0) --> q2
q1 (1) --> q0
q1 (0) --> q3
q1 (epsilon) --> q_accept
q2 (0) --> q2
q2 (1) --> q3
q2 (epsilon) --> q_accept
q3 (Sigma) --> q3

2. Eliminate State q3

Resulting transitions:
q1 to q_accept remains: epsilon
q2 to q_accept remains: epsilon

3. Eliminate State q2

Remaining transitions:
q0 to q1 on 1
q1 to q0 on 1
q1 to q_accept on epsilon

4. Eliminate State q1

New loop q0 to q0:
q0 to q1 (1) followed by q1 loop followed by q1 to q0 (1)
R_0,0 = 1 (empty set)* 1

R_0,0 = 11

New path q0 to q_accept:
R_0,accept (old) OR R_0,1 R_1,1* R_1,accept
R_0,accept = 00* OR 1 (empty set)* epsilon
R_0,accept = 00* OR 1

5. Final Regular Expression

Final Regular Expression:
R = (11)* (00* OR 1)