1. Let $M_2$ be the Turing machine whose description is given in Example 3.7. Give the sequence of configurations that $M_1$ enters when started on the following input strings.
   (a) 0000.
   (b) 00000.

2. Exercise 3.7.

3. Give an implementation-level description of a Turing machine that decides the language
   \[ \{0^m1^p2^r | m > p \text{ and } r > p\} \].


5. Problem 3.11.

6. A deterministic JFLAP Turing machine differs from our definition in several ways.
   - A deterministic JFLAP TM can have more than one accept state. (No further moves are allowed once an accept state is reached.)
   - A JFLAP TM does not have any reject states.
   - A deterministic JFLAP TM can get stuck (i.e., reach a state and tape symbol combination where there is no next move even though the state is not an accept state).

   For a JFLAP TM, an accepting computation is one that reaches an accept state, while a rejecting computation is one that gets stuck (in a non-accept state).

   Show how to transform a deterministic JFLAP TM into an equivalent deterministic TM as we have defined this concept.

7. Problem 3.15 d, e.

8. Problem 3.16 b, d.

9. If $A$ is a language, then $PREFIX(A)$ is the language
   \[ \{u|uv \in A \text{ for some string } v\} \]
   (a) Prove that if $A$ is decidable, then $PREFIX(A)$ is Turing recognizable.
(b) Prove that if $A$ is Turing recognizable, then $PREFIX(A)$ is Turing recognizable.

[Since every decidable language is Turing recognizable, this part implies the first part, but since the proof is harder, I made it a separate part.]