Questions 3, 4, and 5 consider a very simple computer called the Plum. Plum has a finite set \( Q \) of states, two registers \( R \) and \( S \), and a read-only input tape containing symbols in \( \Sigma \). Each register can hold a nonnegative integer, which can be arbitrarily large. Plum operates with a specific positive integer \( m \) (which is fixed once and for all when the machine is built), and operates as follows. Let \( \text{rem}(A, m) \) be the remainder when \( A \) is divided by \( m \), and \( \text{quo}(A, m) \) be the quotient when \( A \) is divided by \( m \).

Each transition reads the next character as long as there is unread input. Once all input is read, a special symbol, EOF, is the result of the read. Each transition examines (1) the current state, (2) \( \text{rem}(R, m) \), and (3) the symbol from \( \Sigma \cup \{\text{EOF}\} \) from the read. Based on this, it determines a new state. It also determines an adder \( \alpha \) with \( 0 \leq \alpha < m \), and does one of the following:

- **R**: replaces \( R \) by \( \text{quo}(R, m) \) and \( S \) by \( m \times S + \alpha \), or
- **S**: replaces \( R \) by \( m \times (m \times \text{quo}(R, m) + \alpha) + \text{rem}(S, m) \) and \( S \) by \( \text{quo}(S, m) \).
- **T**: replaces \( R \) by \( m \times \text{quo}(R, m) + \alpha \) and leaves \( S \) unchanged.

Thus the transition function is a mapping

\[
Q \times \{0, \ldots, m-1\} \times (\Sigma \cup \{\text{EOF}\}) \mapsto Q \times \{0, \ldots, m-1\} \times \{R, S, T\}
\]

Initially \( R = S = 0 \). Plum accepts when it enters a designated accept state. You may need to make further assumptions to answer the questions, so state them clearly if you do.

**The Questions:**

1. Draw a state diagram for a Turing machine that decides the language \( \{1^n\#b : b \text{ is the binary representation of } n\} \). (For example, 11111\#101 is in the language, but 111\#110 is not.) Give a brief explanation of your design.

2. Draw a state diagram for a Turing machine that decides the language of palindromes in \( \{a, b\}^* \) that have the same number of \( a \)s and \( b \)s. Give a brief explanation of your design.

3. Show that for every Turing-recognizable language \( L \), there is a Plum computer that recognizes \( L \). (Hint: Can you turn a Turing machine into an equivalent Plum?)

4. Show that if language \( L \) is recognized by a Plum computer, it is Turing-recognizable. (Hint: Can you turn a Plum into an equivalent Turing machine?)

5. **(not to be graded)** Somebody sold me a bad Plum! I can change the transition function, but it only works with \( m = 1 \). What languages can it recognize?