Design Turing machines \( M = (Q, \Sigma, \Gamma, \delta, \text{start}, \text{accept}, \text{reject}) \) for each of the following tasks, either by listing the states \( Q \), the tape alphabet \( \Gamma \), and the transition function \( \delta \) (in a table), or by drawing the corresponding labeled graph.

Each of these machines uses the input alphabet \( \Sigma = \{1, \#\} \); the tape alphabet \( \Gamma \) can be any superset of \( \{1, \#, \Box, \triangleright\} \) where \( \Box \) is the blank symbol and \( \triangleright \) is a special symbol marking the left end of the tape. Each machine should reject any input not in the form specified below.

1. On input \( 1^n \), for any non-negative integer \( n \), write \( 1^n \#1^n \) on the tape and accept.

2. On input \( \#^n1^m \), for any non-negative integers \( m \) and \( n \), write \( 1^m \) on the tape and accept. In other words, delete all the \( \# \)s and shift the \( 1 \)s to the start of the tape.

3. On input \( \#1^n \), for any non-negative integer \( n \), write \( \#1^{2n} \) on the tape and accept. (Hint: Modify the Turing machine from problem 1.)

4. On input \( 1^n \), for any non-negative integer \( n \), write \( 1^{2n} \) on the tape and accept. (Hint: Use the three previous Turing machines as subroutines.)