1. Design a TM to accept \( \{ww^{rev} \mid w \in \{0,1\}^*\} \) (i.e., even-length palindromes)

Here’s one way to do this:

2. Design a Turing Machine to accept the strings that have the same number of 0’s and 1’s,
   a. Start at the beginning of the string; move to the right until you find a 0 or 1. If you get to a blank without finding 0 or 1, accept.
   b1. If you found a 0, replace it with an X and keep moving to the right looking for a 1, then replace the 1 with an X and go back to the beginning of the string and step a.
   b2. If you initially found a 1 replace it with an X, then find and replace the next 0 with an X and go back to the beginning.
   In either step b1 or b2, if you fail to find the digit you seek halt without accepting.

3. Design a TM to accept \( \{ww \mid w \in \{0,1\}^*\} \) You might find non-determinism helpful. It is sufficient to break this into steps that can clearly be handled by a TM; you don’t need to write out all of the states and transitions unless you want to.
   a) Go to the right end of the string
   b) If the last symbol is a 0 overwrite it with a B, move R, overwrite the B with 0, move L
      If the last symbol is a 1 overwrite it with a B, move R, overwrite the B with 1, move L
   c) Repeat step (b) and arbitrary number of times.
   Nondeterministically move R to a B, overwrite it with c, then go to left end of the input
d) The tape is now wcw. Match 0 or 1 before c to 0 or 1 after c. Overwrite both with X
   e) When you go to the start of the input and find that everything before c is X and
everything after c is X, accept the input.

4. Design a TM that starts with the binary code for a number N on its tape and ends with the code for N+1. So if it starts with 10011 it ends with 10100 and if it starts with 1111 it ends with 10000.

5. Here is a non-deterministic TM. Find all configurations that can be derived from A011

A011=>1A11=>10B1=>101B<blank>=>101<blank>C
=>1A01
=> 11A1
=> 110B<blank>110<blank>C