Problem 1: Define the following words, phrases and symbols.

1. Turing Machine
2. \((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\)
3. (Turing) Decidable Language
4. (Turing) Recognizable Language
5. (Recursively) Enumerable Language
6. Looping / Not Halting

Problem 2: Mark each of the following statements either true or false.

1. A Turing machine has a single start state, but may have many accept states.
2. It is possible to make a Turing machine with only one state.
3. A Turing machine halts when its head reaches the end of its input.
4. All decidable languages are regular languages.
5. A nondeterministic TM with \(k\)-heads can recognize more languages than a deterministic TM with \(k\)-tapes.
6. A Turing machine might not halt on a finite input string.
7. A language \(L\) can be both co-decidable and undecidable.

Problem 3: Describe the operation of a basic Turing machine that recognizes the language \(L = \{ww^R : w \in \{0, 1\}^*\}\). Explain how the Turing machine head(s) move and mark the tape(s), without listing the specific details of each transition. (This is what Sipser calls an “implementation description” on p. 145.)

How would you recognize the language \(\{0^n10^{2n}10^{3n} : n \geq 1\}\) using a 3-headed, single tape TM?

Problem 4: Show that the class of decidable languages is closed under concatenation.

Problem 5: [A Different Turing Machine Model] A Turing Machine (TM) with doubly infinite tape is similar to an ordinary TM except that its tape is infinite to the right as well as left. The tape is initially filled with blanks except for the portion that contains the input. Computation is defined as usual except that the head never encounters an end to the tape as it moves leftward. Show that this type of TM recognizes the class of Turing-recognizable languages.

Problem 6: Robustness of the Turing Machine model
Consider a Turing machine model that uses a 2-dimensional tape, corresponding to the upper right quadrant of the plane. The head of such a Turing machine could move to the right, left, up or down. Sketch a proof that
such a model does not add extra computing power; that is, the class of languages recognized by such Turing machines is the same as the class recognized by basic Turing machines.

**Solution 6:** *The key idea is to notice that at any point of time, the TM has looked at only a finite region of the two-dimensional tape.*

Say the input of the machine is in the first row of the tape, initially the tape head is at the square $(0, 0)$ and all the other tape squares are blank.

Simulate the 2D tape with a 1D tape as follows:

1. The rows of the 2D tape starting from the “bottom-most” row are written in the 1D tape next to each other, as in the figure below. Note that this is indeed possible, because at any point of time, the TM has looked at only a finite number of rows, and a finite number of tape squares in each row.

2. When the machine tries to move right from a square in the 2D tape, we can simulate it in the 1D tape trivially, except when it tries to move right from the rightmost square in a row. In that case, we “make room” for the new tape square by moving all the symbols to the right. For instance, when the machine tries to move right after reading $x_7$ in the figure, we simulate the effect in the 1D tape as follows: We move all the symbols starting from the $\$$ right after $x_7$ in the 1D tape, one square to the right. In the newly vacant square, we write a blank, and continue simulating.

3. When the machine tries to move left from the leftmost square in a row, we do nothing (*The machine hits the wall*). Otherwise, we can trivially simulate it in the 1D tape.

4. When the machine tries to move up, we move left until we reach a $\$$ symbol to determine which column we are in. Then we move right beyond the next $\$$ and into the corresponding column one row up. (To count, we might use an auxiliary tape and later use the 2-tape TM to 1-tape TM conversion). If there is not enough room in the next row, we make room by moving all the tape squares to the right, just as in step 2.

5. The machine tries to move down. We proceed exactly as in Step 4, except that now, we move one row down. In case we are in the bottommost row, we do nothing. (*Again, the machine hits the wall*).
The configuration of the 2D tape at some point of time

The representation of the 2D configuration in the 1D tape