Problem 1. Prove that if a partially decidable language A is m-reducible to its complement $\bar{A}$, then A is decidable.

Problem 2. Prove that every decidable language is m-reducible to the language $L = \{0^*1^*\}$ where the alphabet is $\Sigma = \{0, 1\}$.

Problem 3. Let M_1, M_2 be Turing machines. Is language $\{(M_1, M_2);\ L(M_1) \cap L(M_2) = \emptyset\}$ decidable?

Problem 4. Is language $\{M;\ M \notin L(M)\}$ is (partially) decidable? Note that we encoded every Turing machine by a number and also words are encoded as a number. Therefore, a number M encodes a word (input) and also a Turing machine. So, $M \notin L(M)$ meas that a word M is not accepted by a Turing machine M.

Problem 5. Consider languages L_1 and L_2 where L_1 is undecidable. Is the intersection $L_1 \cap L_2$ necessarily undecidable? Is the union $L_1 \cup L_2$ necessarily undecidable?

Problem 6. Is the following language (partially) deciable?

$$\{(M, w);\ M \text{ terminates on input w and the tape of } M \text{ is empty after the computation}\}$$

Problem 7. Are the following languages (partially) decidable where M is a (code of a) Turing machine and $k \in \mathbb{N}$ and $|w|$ is the length of a word w.

1. $\{w;\ |w| \leq k\}$
2. $\{(M, k);\ |L(M)| \leq k\}$
3. $\{M;\ |L(M)| \geq 10\}$
4. $\{M;\ L(M) = \{w;\ |w| = 10\}\}$
5. $\{(M, k);\ L(M)$ contains a word of length $k\}$

Problem 8. Are the following problem (partially) decidable?

1. For a given Turing machine M, a state q of M and an input w, does M enter the state q during computation $M(w)$?
2. For a given Turing machine M and a state q of M, is there an input w such that M enter the state q during computation $M(w)$?
3. For a given Turing machine M and a state q of M, does M enter the state q during computation $M(w)$ for every word w?

Problem 9. A deterministic queue automaton (DQA) is defined the same way as a deterministic pushdown automata (DPDA), except that it has a queue instead of a stack. In other words, a DQA is a deterministic finite automaton augmented with an unbounded queue, together with the operations of (a) pushing a symbol onto the "back" of the queue, and (b) popping the symbol at the "front" of the queue. Show that DQAs are equivalent in power to Turing machines: that is, any given language L is decidable by a DQA if and only if it's decidable by a Turing machine.