Exercise 1: Constructing Turing Machines  

(a) $L_1 = \{a^ib^ia^jb^j|i, j > 0\}$

(b) Language $L_2$ of all strings over alphabet $\{a, b\}$ with the same number of $a$'s and $b$'s.

Remark: It is sufficient to give a detailed description of the Turing Machines. You do not need to give formal definitions.

Exercise 2: Semi-Decidable vs. Recursively Enumerable (3+3 Points)

Very often people in computer science use the terms *semi-decidable* and *recursively enumerable* equivalently. The following exercise shows in which way they actually are equivalent. We first recall the definition of both terms.

A language $L$ is *semi-decidable* if there is a Turing machine which accepts every $w \in L$ and does not accept any $w \notin L$ (this means the TM can either reject $w \notin L$ or simply not stop for $w \notin L$).

A language is *recursively enumerable* if there is a Turing machine which eventually outputs every word $w \in L$ and never outputs a word $w \notin L$.

(a) Show that any recursively enumerable language is semi-decidable.

(b) Show that any semi-decidable language is recursively enumerable.

Exercise 3: Halting Problem (2+2+2+2 Points)

The *special halting problem* is defined as

$$H_s = \{\langle M \rangle | \langle M \rangle \text{ encodes a TM and } M \text{ halts on } \langle M \rangle \}.$$ 

(a) Show that $H_s$ is undecidable.

*Hint: Assume that $M$ is a TM which decides $H_s$ and then construct a TM which halts iff $M$ does not halt. Use this construction to find a contradiction.*

(b) Show that the special halting problem is recursively enumerable.

(c) Show that the complement of the special halting problem is not recursively enumerable.

*Hint: What can you say about a language $L$ if $L$ and its complement are recursively enumerable? (if you make some observation for this, also prove it)*

(d) Let $L_1$ and $L_2$ be recursively enumerable languages. Is $L_1 \setminus L_2$ recursively enumerable as well?

(e) Is $L = \{w \in H_s | |w| \leq 1742\}$ decidable? Explain your answer!