Exercise 1

Task 1. Given the alphabet $\Sigma = \{0, 1\}$, construct a Turing machine that decides the language $L = \{w \in \Sigma^* : |w|_0 = |w|_1\}$ of all words over Σ with same number of 0's and 1's.

Task 2. In the definition of the class **P** one can replace the Turing machine model by more practical models of computation. Describe how a Turing machine (TM) can simulate a random-access machine (RAM) as defined by the following simple machine model.

The machine has a memory formed by cells, each storing a natural number m_i and indexed by a natural number i. A program is a sequence $I_0, I_1, \ldots, I_n$ of the following instructions.

- `SET`(i, a): assign $m_i \leftarrow a$ for some constant a.
- `MOV`(i, j): assign $m_i \leftarrow m_j$.
- `ADD`(i, j): assign $m_i \leftarrow m_i + m_j$.
- `SUB`(i, j): assign $m_i \leftarrow m_i \dot{-} m_j \coloneqq \max\{m_i - m_j, 0\}$.
- `JZ`(i, l): if $m_i = 0$, then continue execution with I_l for some constant l.

For every instruction, i and j can be a fixed address k (representing the cell storing m_k) as well as an indirect address $*k$ (representing the cell that stores m_{m_k}). The machine begins executing I_0, and after executing I_l it continues with I_{l+1} except possibly for `JZ`. The input and output are stored in agreed-upon memory cells. A non-accessed cell contains the value zero. Execution terminates upon reaching the first non-existent instruction.

Hint: In the lecture we defined single-tape Turing machines, but you may also use any constant number of tapes because a single-tape TM can simulate a k-tape TM.

Task 3. Consider the following probability amplification method (slide 13) to improve the error probability of a probabilistic Turing machine (PTM) M.

- Choose strings $v_1, ..., v_k \in \{0, 1\}^{r(n)}$ independently and uniformly at random.
- Accept the input u if $\langle u, v_i \rangle \in L(M)$ for at least $k/2$ many $i \in [1, k]$

Use the Chernoff bound (or Hoeffding's inequality) to show that this algorithm has an error probability that is bounded above by $2^{-\Omega(k)}$.

Task 4. Let L_1 and L_2 be languages belonging to **NP**.

(a) Show that $L_1 \cup L_2$ and $L_1 \cap L_2$ also belong to **NP**.

Now suppose additionally that $L_1 \leq L_2$.

(b) Show that if L_2 is in **P**, then so is L_1. What can you say about the converse?

(c) Show that if L_1 is **NP**-complete, then so is L_2. Does the converse hold?