1. Construct a Turing Machine $M$ that decides $L = \{a^ib^jc^kd^\ell : i + k = j + \ell\}$ (ie $M$ stops in state “YES” if the input is in $L$ and “NO” if it isn’t). Explain how your Turing Machine works.

2. Construct a k-string Turing Machine $M$ which starts with a number in unary on the first tape and ends with the same number in binary on the last tape. In other words, if $M$ is started with a string of $n$ 1’s on the first tape then $M$ should halt with the number $n$ in binary on the last tape. Explain how your Turing Machine works.

3. In the proof that HALTING is undecidable, the (assumed to exist) Turing Machine $M_H$ is modified to a new Turing Machine $D$. Assume that $M_H$ is represented by the image below, draw the Turing Machine $D$.

4. Let $\text{REG} = \{M : L(M) \text{ is a regular language}\}$. Prove that $\text{REG}$ is not recursive by reducing H to REG. For those preferring the “question” version: Given a Turing Machine $M$, does $M$ accept a regular language?