1. Write two accepting computation sequences for the input string “aaaaaa” (or $a^6$) using this non-deterministic Turing machine: **[12 points]**

   $1aaaaaa \rightarrow b2aaaaa \rightarrow bc1aaaa \rightarrow bcb2aaa \rightarrow bc1bca \rightarrow bcbcb2a \rightarrow bc1bcb1B \rightarrow bc1bcb4c.$

   $1aaaaaa \rightarrow b2aaaaa \rightarrow bd3aaaa \rightarrow bde1aaa \rightarrow bdeb2aa \rightarrow bdeb3a \rightarrow bdebde1B \rightarrow bdebde4e.$

2. Draw a deterministic Turing machine that accepts {strings over alphabet {a,b} with number of a’s equal to number of b’s}. Example: aaababbbabaa, which has 7 a’s and 7 b’s. **[12 points]**
3. Draw a deterministic Turing machine that accepts \( \{ \text{palindromes over alphabet \{a,b\}} \} \). Recall that string \( w \) is a palindrome if \( w = w^R \), where \( w^R \) denotes the reverse of string \( w \). Examples: abaaaba or babaabab. [12 points]

4. Draw a deterministic Turing machine that accepts \( \{a^n b^n c^n d^n\} \). Example: aaabbbcccddd. [12 points]
5. Define these terms: [12 points]

a. Decidable (or recursive)

A language that is accepted by a Turing machine which does halt on every input string.

b. Turing-recognizable (or recursively enumerable)

A language that is accepted by a Turing machine which might not halt for some input strings.

c. Church’s thesis

If a problem is solvable using any model of computation, then the problem is solvable on a Turing machine. This statement is a widely-believed conjecture that cannot be formally proven.

d. Halting problem

Given a program and an input string, will the program eventually halt for this input string? The halting problem is both Turing-recognizable and undecidable.

e. Universal language

Given a program and an input string, will the program accept this input string? The universal language is both Turing-recognizable and undecidable.

f. Post’s correspondence problem

Given a list of pairs of strings \([\alpha_1, \beta_1], \ldots, [\alpha_n, \beta_n]\), does there exist a sequence of indices \(j_1, \ldots, j_m\) with \(m \geq 1\) such that \(\alpha_{j_1} \ldots \alpha_{j_m} = \beta_{j_1} \ldots \beta_{j_m}\)? This problem is both Turing-recognizable and undecidable.