1. (a) Give a regular expression that describes the language
\[ \{ w \in \{0,1\}^* \mid \text{w begins and ends with a different symbol} \} \].
8p.

(b) Design a deterministic finite automaton that recognises the language in part (a).
7p.

2. (a) Design a context-free grammar for the language
\[ L = \{ w \in \{a,b\}^* \mid \text{w contains equally many a’s and b’s} \} \].

Draw the corresponding parse trees for the sentences aabb, abab and baab.
5p.

(b) Is the grammar you designed in part (a) unambiguous or ambiguous? (Justify your answer.)
5p.

(c) Prove (precisely!) that the language L in part (a) is not regular.
5p.

3. Design a deterministic single-tape Turing machine that recognises (“decides”) the language L considered in the previous problem. (Present the Turing machine preferably as a state diagram rather than a transition table.) Show the computation sequences (“runs”) of your machine on inputs abab and bba.
15p.

4. (a) Define the notions of a recursive (“decidable”) and recursively enumerable (“semi-decidable”) language, and explain their relation to issues in computer programming.
8p.

(b) Give an example of a language that is recursively enumerable, but not recursive. (You should provide a precise definition for the language, but need not prove any of its claimed properties.) Explain the significance of your example from the point of view of computer programming.
7p.

Total 60p.