

- N.B.** (1) Question No. 1 is **compulsory**.  
 (2) Attempt any **four** questions from remaining **six** questions.  
 (3) **Figures** to the **right** indicate **full** marks given to the question.  
 (4) Assume **suitable** data, if **necessary**.

1. (a) Distinguish between NFA and DFA. 3  
 (b) Design a FSM to determine whether a binary number is divisible by 3. 5  
 (c) Design a Mealy Machine to convert HEX number to OCTAL number. 6  
 (d) Design a PDA to accept  $a^n b^n$  over an alphabet  $\Sigma = \{a, b\}$ . 6
  
2. (a) Design a TM to perform  $C = A + B$  for binary numbers when the tape has input  $\$ A + B \$$  and output should be  $\$ C \$$ . 10  
 (b) Construct DFA accepting the following language :— 10
  - (i) String ending with 111 or 110.
  - (ii) String containing even number of 0's and odd number of 1's.
  
3. (a) Explain GNF and CNF. Convert the following grammar into GNF and CNF 10  

$$S \rightarrow ABA$$

$$A \rightarrow aA^{|}$$

$$B \rightarrow bB^{|}$$
 (b) Let G be the grammar 10  

$$S \rightarrow aB^{|}bA$$

$$A \rightarrow a^{|}aS^{|}bAA$$

$$B \rightarrow b^{|}bS^{|}aBB$$

Find leftmost derivation, rightmost derivation and parse tree for the string "bbaaabbaba". 10
  
4. (a) Design a PDA to accept a string containing equal number of 0's and 1's. 10  
 (b) State and explain pumping Lemma for regular languages. Using Pumping lemma, show that following grammars are not regular. 10
  - (i)  $L = \{ a^n b^n c^n \mid n > 0 \}$
  - (ii)  $L = \{ SS \mid S \in \{a,b\}^* \}$
  
5. (a) Construct  $\epsilon$ -NFA from a regular expression  $((ab)^* + bb)^*$  and convert into minimized DFA. 10  
 (b) Explain with examples the classification of various grammars. 10
  
6. (a) Design a suitable parser for "Balanced begin end blocks". Let begin be represented by 'b' and end by 'e'. Assume suitable terminals and non terminal as required. 10  
 (b) Design a TM to find GCD of two unary numbers. 10
  
7. Write short notes on (any **four**) :— 20
  - (a) Power of UTM
  - (b) Halting Problem
  - (c) Operator precedence parser
  - (d) Moore Machine
  - (e) NP-Hard and NP- Complete problem
  - (f) Power of NPDA.