	ITER, SIKSHA 'O' ANUSANDHAN (Deemed to be University)		Assignment
Branch	**CSE/CSIT**	Programme	**B.Tech**
Course Name	**Introduction to the Theory of Computation**	Semester	**5th**
Course Code	**CSE3731**	Academic Year	**2024-25**

ASSIGNMENT - II

Submission due date: 17/12/2024

Learning Level (LL)			
	L1: Remembering	**L3:** Applying	**L5**: Evaluating
	L2: Understanding	**L4**: Analysing	**L6**: Creating

Q's	Questions	COs	LL
1	a) If G= ({S,A}, {a,b},P,S) where P consists of {**S→aAS|aSS|ε, A→SbA|ba**}. Show that the grammar G is ambiguous for the string **aabaa** b) Let the context-free grammar G be given by the productions {**S →AB, A →a, B →b, B →C, E →c**}. Find the context-free grammar G1 such that every variable in G1 derives some terminal string. c) Construct a grammar in Chomsky normal form(CNF) from the equivalent CFG, G, with production rules **S → aAbB, A →aA|a, B →bB|b**.	**CO3**	**L3**
2	a) Explain Chomsky Hierarchy and show the relationship between the different types of grammars. b) Using pumping lemma prove that the language L= **{ a^n | n is a prime**} is not a Context-free language.	**CO3**	**L4**
3	a) Design a PDA that recognizes the Language L= {**$a^i b^j c^k$ | i,j,k ≥ 0 and i=j or i=k**}. Show the stack implementation for the string **aaabbccc**. b) Construct the PDA that recognizes the Language L= {**ww^R| w ϵ {0,1}***}. Show the stack implementation for the string **10100101**.	**CO3**	**L3,L6**
4	a) Construct a Turing Machine that recognizes the language L={ n^{2^n} **| n>=0**} b) Construct a TM that decides the language L= {**w#w | w ϵ {0,1}***}	**CO4**	**L3,L6**
5	a) Show that halting problem of a Turing Machine is un-decidable. b) What is a P, NP, NP-Complete and NP-Hard problem? Explain the relationship of these concepts with the help of a Venn-diagram and give some examples for each problem.	**CO5,CO6**	**L5**

Course Outcomes	By the end of the course, through lectures, readings, home works, assignments, and exams, students will be able to:	
	CO1	Enhance/develop ability to understand and conduct mathematical proofs for computation and algorithms.
	CO2	Design and analyze finite automata and regular expression for describing regular languages.
	CO3	Design and analyze pushdown automata, and context-free grammars.
	CO4	Design and analyze Turing machines.
	CO5	Enhance the ability to understand the decidability, undecidability, and reducibility criteria of various computational problems.
	CO6	Demonstrate the understanding of key notions, such as algorithm, computability and complexity through problem solving.

- ✓ ***Assignment scores/markings depend on neatness and clarity.***
- ✓ ***Plagiarized assignments will be given a zero mark.***
- ✓ ***Submit the hard binding copy of your assignment by the due date, i.e. 17.12.2024***
- ✓ ***Submit the assignment handwritten on A4 size papers and spirally bound to your ITC class teacher. A front page must be present containing the details of the subject, the assignment and the student.***