

GANPAT UNIVERSITY



U. V. PATEL COLLEGE OF ENGINEERING

PRACTICAL LIST

2CEIT601: Theory of Computation

**Computer Engineering/ Information
Technology/ Computer Engineering with
Artificial Intelligence**

A.Y. 2023-2024(Even Session)

GANPAT UNIVERSITY
U.V. PATEL COLLEGE OF ENGINEERING
B. Tech Semester VI
Computer Engineering/ Information Technology/ Computer
Engineering with Artificial Intelligence
2CEIT601: Theory of Computation
List of Experiments

Tools – Telnet

Sr. No	Experiments	CO Mapping
1	<p>Introduction to LEX Programming Environment for formal language.</p> <ul style="list-style-type: none"> ● Introduction of “VI” Editor with Commands. ● Introduction of Lex. ● Explain How to Compile Lex program with sample program. 	1
2	<p>Write a separate Lex program for following:</p> <ol style="list-style-type: none"> 1. Write RE that accepts zero or one (at most one) occurrence of ‘a’ over the alphabets {a,b}. 2. Write RE that accepts either ‘a’ or ‘b’. 3. Write RE that accept either ‘a’ or ‘b’ or ‘c’ without using . 4. Write RE that accepts zero or more occurrences of ‘a’ and single occurrences of ‘b’. 5. Write RE that accepts all the strings which ends with ‘b’. 6. Write RE for a new line. 7. Write RE that accepts ‘\n’. 8. Write a RE that accepts any character except ‘\’ and ‘n’. 9. Write all the strings which are accepted by [a b c*]. 10. Write a RE that accepts any character except ‘a’ and ‘b’. 	1,2
3	<p>Write a separate Lex program for following:</p> <ol style="list-style-type: none"> 1. Write more than one RE that accepts string ‘abc’. 2. Is there any difference between ‘abc’ and “abc”? Justify your answer. 3. Which are the strings accepted by (“abc”)*. 4. Write the RE that accepts zero or more occurrences of digit and capital letters. 5. Write valid and invalid strings accepted by following regular expressions. <ol style="list-style-type: none"> I. (a-b)?[0-9 A-Z]* II. [^ab][0-9]* III. ^[ab][0-9 A-Z]+ IV. [0-9][A-Z]\$ V. [A-Z a-z]{6} VI. [a+b]{6} VII. [a+b] 	2,3

4	<ol style="list-style-type: none"> 1. Demonstrate the use of Lex predefined variables (yytext, yyleng, yyin) with the help of a program. 2. Write a Lex program to recognize character, string and special symbols from given input. 3. Write a Lex program to validate mobile numbers. (i.e Number having length of 10 is valid) 4. Write a Lex program to differentiate mobile number and landline number. (Apply following constraints: Both mobile and landline numbers have 10 digits but landline number starts with digit 0 to 6 and mobile number has starting digit 7 to 9.) 	2,3
5	<ol style="list-style-type: none"> 1. Write a Lex program to recognize identifiers in C, <ul style="list-style-type: none"> • C identifiers should have following constraint • It should start with either letter or underscore (_) sign. • It should not contain special symbols. 2. Write a Lex program for validation of Email-Add. (Consider Email Add. from any domain e.g. @gmail.com) 3. Write a Lex program to identify integer, float and exponential value. Examples: <ul style="list-style-type: none"> 123 Integer 12.23 Float 12E23 Exponential 12.25E23 Exponential 12.25E-25 Exponential -12.25E-25 Exponential 12.25E25.25 Other (Reason: float value after E not allowed) 	3
6	<ol style="list-style-type: none"> 1. Write a Lex program to count the number of words, characters and lines from user input. 2. Write a Lex program to count the number of words, characters and lines from a file. 3. Write a Lex program to convert lower case letter to upper case from user input and terminate the program if user enters 0. 4. Write a Lex program to convert the lowercase letter of a given file to upper case. 5. Write a Lex program to check whether the IP address entered by the user is valid or not. 6. Write a Lex program to validate IP addresses from user specified files. Consider the content of ipaddress.txt file as below: 192.168.2.255 123.256.89.89 Hello 0.0.0.289 	3

	<p>0.0.0.0. 289.255.245.243 0.0.0.214 10.0.0.245 0.0.0.0 255.255.255.255</p> <p>Expected O/P: 192.168.2.255 – Valid IP address 123.256.89.89 - Invalid IP address Hello - Invalid IP address 0.0.0.289 - Invalid IP address 0.0.0.0. - Invalid IP address 289.255.245.243 - Invalid IP address 0.0.0.214 - Valid IP address 10.0.0.245 - Valid IP address 0.0.0.0 - Valid IP address 255.255.255.255 - Valid IP address</p> <p>7. Write a Lex program to check whether the Date entered by the user in format (dd/mm/yyyy) is valid or not.</p> <p>Note: Consider the date range from 1 to 31 for every month and year from 1900 to 2999</p>	
7	<p>Write a regular expression for following languages and use it in Lex program</p> <ol style="list-style-type: none"> 1. The language of all strings contains exactly two 0's 2. The language of all strings contains at least two 0's 3. The language of all strings ending in 1 and not containing 00 4. String with odd number of 1's 5. The language of all strings that do not end with 01 6. The language of all string not containing 00 7. The language of all string containing either 10 or 001 	1,2
8	<ol style="list-style-type: none"> 1. Write a program to identify the word and change the case of each character of word (i.e. if character in word is in lowercase then convert it to uppercase and if it is uppercase then convert to lowercase) 2. Write a Lex program to count the number of comments (single line or multiple lines) in a given C source file. 3. Write a Lex program to display your name when 0 is entered and nothing should be displayed on the screen 4. Write a Lex program to count the word "India" from a file. Expected Input from file: India is a great country and India has a large population. Expected output: word count = 2 5. Display content of the text file on the output screen with the word "Hello" replaced by "Hi". 	2,3
9	Design a Program to create PDA machine that accept the well-formed parenthesis	1,3
10	Design a Turing Machine that calculate 2's complement of given binary string.	1,3
11	Study about the time complexity of Deterministic and Non Deterministic Turing Machine	4

Course Outcomes:

CO1	Demonstrate advanced knowledge of formal computation and its relationship to formal languages.
CO2	Distinguish different computing languages and classify their respective types.
CO3	Recognize and comprehend formal reasoning about languages.
CO4	Show a competent understanding of the basic concepts of complexity theory.

Mapping of CO and PO:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3
CO1	2	1	2	1	0	0	0	0	0	0	0	2	1	2	1
CO2	2	2	2	1	0	0	0	0	0	0	0	2	1	2	2
CO3	3	3	1	2	0	0	0	0	0	0	0	1	2	2	2
CO4	2	2	1	2	0	0	0	0	0	0	0	1	3	2	0

Prepared By

Prof. Ritesh Upadhyay
Assistant Professor, CE Dept.

Prof. Amit Solanki
Assistant Professor, CE Dept.

Prof. Chirag Patel
Assistant Professor, CE Dept.