**Department of Computer Science & Engineering (AI&ML)**

**MOBILE APP DEVELOPMENT (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Introduction to mobile technologies and devices, Android platform and applications overview. |
| Exercise-2 | Setting Android development environments. |
| Exercise-3 | Writing Android applications, Understanding anatomy of an Android application. |
| Exercise-4 | Develop an application that uses GUI components, Font and Colors. |
| Exercise-5 | Develop an application that uses Layout Managers and event listeners. |
| Exercise-6 | Write an application that draws basic graphical primitives on the screen. |
| Exercise-7 | Develop an application that makes use of databases. |
| Exercise-8 | Develop an application that makes use of Notification Manager. |
| Exercise-9 | Implement an application that uses Multi-threading. |
| Exercise-10 | Develop a native application that uses GPS location information. |
| Exercise-11 | Implement an application that writes data to the SD card. |
| Exercise-12 | Implement an application that creates an alert upon receiving a message. |
| Exercise-13 | Write a mobile application that makes use of RSS feed. |
| Exercise-14 | Develop a mobile application to send an email. |
| Exercise-15 | Develop a Mobile application for simple needs (Mini Project) |

**Department of Computer Science & Engineering (AI&ML)**

R PROGRAMMING LAB (R20)

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Department of Computer Science & Engineering (AI&ML)**

**INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Implementation of DFS for water jug problem using LISP/PROLOG |
| Exercise-2 | Implementation of BFS for tic-tac-toe problem using LISP/PROLOG/Java |
| Exercise-3 | Implementation of TSP using heuristic approach using Java/LISP/Prolog |
| Exercise-4 | Implementation of Simulated Annealing Algorithm using LISP/PROLOG |
| Exercise-5 | Implementation of Hill-climbing to solve 8- Puzzle Problem |
| Exercise-6 | Implementation of Monkey Banana Problem using LISP/PROLOG Implement and demonstrate FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a . csv file. |
| Exercise-7 | For a given set of training data examples stored in a .csv file, implement and demonstrate the candidate elimination algorithm to output a description of the set of all hypotheses consistent with the training examples |
| Exercise-8 | Write a program to demonstrate the working of the decision tree classifier. Use appropriate dataset for building the decision tree and apply this knowledge to classify a new sample. |
| Exercise-9 | Write a program to demonstrate the working of Decision tree regressor. Use appropriate dataset for decision tree regressor. |
| Exercise-10 | Write a program to demonstrate the working of Random Forest classifier. Use appropriate dataset for Random Forest Classifier |
| Exercise-11 | Write a program to demonstrate the working of Logistic Regression classifier. Use appropriate dataset for Logistic Regression. |

**Department of Computer Science & Engineering (AI&ML)**

**INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (3\*4) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram. |
| Exercise-2 | Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average. |
| Exercise-3 | Write a program that uses a *for* loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89. |
| Exercise-4 | Write a program that asks the user for their name and how many times to print it. The program should print out the user’s name the specified number of times. |
| Exercise-5 | Use a *for* loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.  \*  \*\*  \*\*\*  \*\*\*\* |
| Exercise-6 | Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not. |
| Exercise-7 | Write a program that asks the user for two numbers and prints *Close* if the numbers are within .001 of each other and Not close otherwise. |
| Exercise-8 | Write a program that asks the user to enter a word and prints out whether that word contains any vowels. |
| Exercise-9 | Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters *abcde* and *ABCDE* the program should print out *AaBbCcDdEe*. |
| Exercise-10 | Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be 1,000,000. |
| Exercise-11 | In algebraic expressions, the symbol for multiplication is often left out, as in 3x+4y or 3(x+5). Computers prefer those expressions to include the multiplication symbol, like 3\*x+4\*y or 3\*(x+5). Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate. |
| Exercise-12 | Write a program that generates a list of 20 random numbers between 1 and 100.  (a) Print the list.  (b) Print the average of the elements in the list.  (c) Print the largest and smallest values in the list.  (d) Print the second largest and second smallest entries in the list  (e) Print how many even numbers are in the list. |
| Exercise-13 | Write a program that asks the user for an integer and creates a list that consists of the factors of that integer. |
| Exercise-14 | Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4. |
| Exercise-15 | Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0]. |
| Exercise-16 | Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements,it is much shorter with lists and it is also easier to add new conversions if you use lists. |
| Exercise-17 | Write a function called *sum\_digits* that is given an integer num and returns the sum of the digits of num. |
| Exercise-18 | Write a function called *first\_diff* that is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return -1. |
| Exercise-19 | Write a function called *number\_of\_factors* that takes an integer and returns how many factors the number has. |
| Exercise-20 | Write a function called *is\_sorted* that is given a list and returns True if the list is sorted and False otherwise. |
| Exercise-21 | Write a function called root that is given a number x and an integer n and returns x1/n. In the function definition, set the default value of n to 2. |
| Exercise-22 | Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100. |
| Exercise-23 | Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list.  (a) Do this using the sort method. (b) Do this without using the sort method. |
| Exercise-24 | Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can’t exceed the number of occurrences of the letter in the user’s word. |
| Exercise-25 | Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons. |
| Exercise-26 | Write a program that reads a list of temperatures from a file called *temps.txt*, converts those temperatures to Fahrenheit, and writes the results to a file called ftemps.txt. |
| Exercise-27 | Write a class called Product. The class should have fields called name, amount, and price, holding the product’s name, the number of items of that product in stock, and the regular price of the product. There should be a method *get\_price* that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called *make\_purchase* that receives the number of items to be bought and decreases amount by that much. |
| Exercise-28 | Write a class called Time whose only field is a time in seconds. It should have a method called *convert\_to\_minutes* that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called *convert\_to\_hours* that returns a string of hours, minutes, and seconds formatted analogously to the previous method. |
| Exercise-29 | Write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, c = Converter(9,'inches'). The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the Converter object created above, the user could call c.feet() and should get 0.75 as the result. |
| Exercise-30 | Write a Python class to reverse a string word by word. |
| Exercise-31 | Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox. |
| Exercise-32 | Write a program to demonstrate Try/except/else. |
| Exercise-33 | Write a program to demonstrate try/finally and with/as. |
| Exercise-34 | Write a Python class to implement pow(x, n). |

**Department of Computer Science & Engineering (AI&ML)**

**ALGORITHMS FOR EFFICIENT CODING LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Develop a program and measure the running time for Binary Search with Divide and Conquer |
| Exercise-2 | Develop a program and measure the running time for Merge Sort with Divide and Conquer |
| Exercise-3 | Develop a program and measure the running time for Quick Sort with Divide and Conquer |
| Exercise-4 | Develop a program and measure the running time for estimating minimum-cost spanning Trees with Greedy Method |
| Exercise-5 | Develop a program and measure the running time for estimating Single Source Shortest Paths with Greedy Method |
| Exercise-6 | Develop a program and measure the running time for optimal Binary search trees with Dynamic Programming |
| Exercise-7 | Develop a program and measure the running time for identifying solution for traveling salesperson problem with Dynamic Programming |
| Exercise-8 | Develop a program and measure the running time for identifying solution for 8-Queens problem with Backtracking |
| Exercise-9 | Develop a program and measure the running time for Graph Coloring with Backtracking |
| Exercise-10 | Develop a program and measure the running time to generate solution of Hamiltonian Cycle problem with Backtracking |
| Exercise-11 | Develop a program and measure the running time running time to generate solution of Knapsack problem with Backtracking |

**Department of Computer Science & Engineering (AI&ML)**

**DATA MINING USING PYTHON LAB**  **(R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Demonstrate the following data preprocessing tasks using python libraries. a) Loading the dataset b) Identifying the dependent and independent variables c) Dealing with missing data |
| Exercise-2 | Demonstrate the following data preprocessing tasks using python libraries. a) Dealing with categorical data b) Scaling the features c) Splitting dataset into Training and Testing Sets |
| Exercise-3 | Demonstrate the following Similarity and Dissimilarity Measures using python a) Pearson’s Correlation b) Cosine Similarity c) Jaccard Similarity d) Euclidean Distance e) Manhattan Distance |
| Exercise-4 | Build a model using linear regression algorithm on any dataset |
| Exercise-5 | Build a classification model using Decision Tree algorithm on iris dataset |
| Exercise-6 | Apply Naïve Bayes Classification algorithm on any dataset |
| Exercise-7 | Generate frequent itemsets using Apriori Algorithm in python and also generate association rules for any market basket data. |
| Exercise-8 | Apply K- Means clustering algorithm on any dataset |
| Exercise-9 | Apply Hierarchical Clustering algorithm on any dataset |
| Exercise-10 | Apply DBSCAN clustering algorithm on any dataset. |

**Department of Computer Science & Engineering (AI&ML)**

**DATABASE MANAGEMENT SYSTEMS LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command. |
| Exercise-2 | Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class. |
| Exercise-3 | Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views. |
| Exercise-4 | Queries using Conversion functions (to\_char, to\_number and to\_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next\_day, add\_months, last\_day, months\_between, least, greatest, trunc, round, to\_char, to\_date) |
| Exercise-5 | 1. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)   ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block. |
| Exercise-6 | Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions |
| Exercise-7 | Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR. |
| Exercise-8 | Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES. |
| Exercise-9 | Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions. |

**Department of Computer Science & Engineering (AI&ML)**

**WEB APPLICATION DEVELOPMENT LAB** **(R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Authentication using Java Servlet |
| Exercise-2 | Authentication using JSP |
| Exercise-3 | Connect MySQL database using JSP |
| Exercise-4 | Design and development of Online Book Shop using JSP/Node.js & React.js |
| Exercise-5 | Design and development of Online Examination using JSP/Node.js & React.js |
| Exercise-6 | Design and development of online ticket reservation system using JSP/Node.js & React.js |
| Exercise-7 | Design and development of online library using JSP/Node.js & React.js |
| Exercise-8 | Design and development of online banking using JSP/Node.js & React.js |
| Exercise-9 | Design and development of online job portal using JSP/Node.js & React.js |
| Exercise-10 | Design and development of Online Auction using JSP/Node.js & React.js |

**Department of Computer Science & Engineering (AI&ML)**

**DEEP LEARNING WITH TENSORFLOW** **(R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Implement multilayer perceptron algorithm for MNIST Hand written Digit Classification. |
| Exercise-2 | Design a neural network for classifying movie reviews (Binary Classification) using IMDB dataset. |
| Exercise-3 | Design a neural Network for classifying news wires (Multi class classification) using Reuters dataset. |
| Exercise-4 | Design a neural network for predicting house prices using Boston Housing Price dataset. |
| Exercise-5 | Build a Convolution Neural Network for MNIST Hand written Digit Classification. |
| Exercise-6 | Build a Convolution Neural Network for simple image (dogs and Cats) Classification |
| Exercise-7 | Use a pre-trained convolution neural network (VGG16) for image classification. |
| Exercise-8 | Implement one hot encoding of words or characters. |
| Exercise-9 | Implement word embeddings for IMDB dataset. |
| Exercise-10 | Implement a Recurrent Neural Network for IMDB movie review classification problem. |

**Department of Computer Science & Engineering (AI&ML)**

**MACHINE Learning Lab (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file. |
| Exercise-2 | For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples. |
| Exercise-3 | Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample. |
| Exercise-4 | Exercises to solve the real-world problems using the following machine learning methods: a) Linear Regression b) Logistic Regression c) Binary Classifier |
| Exercise-5 | Develop a program for Bias, Variance, Remove duplicates , Cross Validation |
| Exercise-6 | Write a program to implement Categorical Encoding, One-hot Encoding |
| Exercise-7 | Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets. |
| Exercise-8 | Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. |
| Exercise-9 | Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs |
| Exercise-10 | Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set. |
| Exercise-11 | Apply EM algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program. |

**Department of Computer Science & Engineering (AI&ML)**

**NATURAL LANGUAGE PROCESSING WITH PYTHON (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Demonstrate Noise Removal for any textual data and remove regular expression pattern such as hash tag from textual data |
| Exercise-2 | Perform lemmatization and stemming using python library nltk |
| Exercise-3 | Demonstrate object standardization such as replace social media slangs from a text |
| Exercise-4 | Perform part of speech tagging on any textual data |
| Exercise-5 | Implement topic modeling using Latent Dirichlet Allocation (LDA ) in python |
| Exercise-6 | Demonstrate Term Frequency – Inverse Document Frequency (TF – IDF) using python |
| Exercise-7 | Demonstrate word embeddings using word2vec. |
| Exercise-8 | Implement Text classification using naïve bayes classifier and text blob library |
| Exercise-9 | Apply support vector machine for text classification |
| Exercise-10 | Convert text to vectors (using term frequency) and apply cosine similarity to provide closeness among two text |
| Exercise-11 | . Case study 1: Identify the sentiment of tweets In this problem, you are provided with tweet data to predict sentiment on electronic products of netizens. |
| Exercise-12 | Case study 2: Detect hate speech in tweets. The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets. |

**Department of Computer Science & Engineering (AI&ML)**

**OBJECT ORIENTED PROGRAMMING WITH JAVA LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | **(Basics**) : a) Write a JAVA program to display default value of all primitive data type of JAVA b) Write a java program that display the roots of a quadratic equation ax2+bx=0. Calculate the discriminate D and basing on value of D, describe the nature of root. c) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers. |
| Exercise-2 | (**Operations, Expressions, Control-flow, Strings**) : a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism. b) Write a JAVA program to sort for an element in a given list of elements using bubble sort c) Write a JAVA program to sort for an element in a given list of elements using merge sort. d) Write a JAVA program using StringBuffer to delete, remove character. |
| Exercise-3 | (**Class, Objects**) : a) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method. b) Write a JAVA program to implement constructor. |
| Exercise-4 | **(Methods**) : a) Write a JAVA program to implement constructor overloading. b) Write a JAVA program implement method overloading. |
| Exercise-5 | (Inheritance): a) Write a JAVA program to implement Single Inheritance b) Write a JAVA program to implement multi level Inheritance c) Write a java program for abstract class to find areas of different shapes |
| Exercise-6 | (**Inheritance – Continued**): a) Write a JAVA program give example for “super” keyword. b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved? |
| Exercise-7 | (**Exception**): a) Write a JAVA program that describes exception handling mechanism b) Write a JAVA program Illustrating Multiple catch clauses |
| Exercise-8 | **(Runtime Polymorphism**) : a) Write a JAVA program that implements Runtime polymorphism b) Write a Case study on run time polymorphism, inheritance that implements in above problem |
| Exercise-9 | (**User defined Exception**) : a) Write a JAVA program for creation of Illustrating throw b) Write a JAVA program for creation of Illustrating finally c) Write a JAVA program for creation of Java Built-in Exceptions d) d)Write a JAVA program for creation of User Defined Exception |
| Exercise-10 | **(Threads**): a) Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds ,(Repeat the same by implementing Runnable) b) Write a program illustrating isAlive and join () c) Write a Program illustrating Daemon Threads. |
| Exercise-11 | (**Threads continuity**) : a) Write a JAVA program Producer Consumer Problem b) Write a case study on thread Synchronization after solving the above producer consumer problem |
| Exercise-12 | **(Packages**) : a) Write a JAVA program illustrate class path b) Write a case study on including in class path in your os environment of your package. c) Write a JAVA program that import and use the defined your package in the previous Problem |
| Exercise-13 | (**Applet**) : a) Write a JAVA program to paint like paint brush in applet. b) Write a JAVA program to display analog clock using Applet. c) Write a JAVA program to create different shapes and fill colors using Applet |
| Exercise-14 | (**Event Handling**) : a) Write a JAVA program that display the x and y position of the cursor movement using Mouse. b) Write a JAVA program that identifies key-up key-down event user entering text in a Applet |

**Department of Computer Science & Engineering (AI&ML)**

**OPERATING SYSTEMS & COMPILER DESIGN LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Simulate the following CPU scheduling algorithms: a. Round Robin (b) SJF (c) FCFS (d) Priority |
| Exercise-2 | Simulate the following: a) Multiprogramming with a fixed number of tasks (MFT) b) Multiprogramming with a variable number of tasks (MVT) |
| Exercise-3 | Simulate the following page replacement algorithms: a) FIFO b) LRU c) LFU |
| Exercise-4 | Write a C program that illustrates two processes communicating using shared memory |
| Exercise-5 | Write a C program to simulate producer and consumer problem using semaphores |
| Exercise-6 | Simulate Bankers Algorithm for Dead Lock Avoidance |
| Exercise-7 | Simulate Bankers Algorithm for Dead Lock Prevention |
| Exercise-8 | Write a C program to identify different types of Tokens in a given Program. |
| Exercise-9 | Write a Lex Program to implement a Lexical Analyzer using Lex tool. |
| Exercise-10 | Write a C program to Simulate Lexical Analyzer to validating a given input String. |
| Exercise-11 | Write a C program to implement the Brute force technique of Top down Parsing. |
| Exercise-12 | Write a C program to implement a Recursive Descent Parser |
| Exercise-13 | Write C program to compute the First and Follow Sets for the given Grammar. |
| Exercise-14 | Write a C program for eliminating the left recursion and left factoring of a given grammar |
| Exercise-15 | Write a C program to check the validity of input string using Predictive Parser |
| Exercise-16 | Write a C program for implementation of a Shift Reduce Parser using Stack Data Structure to accept a given input string of a given grammar |
| Exercise-17 | Simulate the calculator using LEX and YACC tool. |