1	Implement depth first search algorithm and Breadth First Search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.
2	Implement Greedy search algorithm for following application • Selection Sort • Prim's Minimal Spanning Tree Algorithm
3	Implement Greedy search algorithm for following application • Selection Sort • Kruskal's Minimal Spanning Tree Algorithm
4	Implement Greedy search algorithm for following application • Selection Sort • Dijkstra's Minimal Spanning Tree Algorithm
5	Implement A star (A*) Algorithm for any game search problem.
6	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound for a graph coloring problem.
7	Implement a solution for a Constraint Satisfaction Problem using Backtracking for a graph coloring problem.
8	Develop an elementary chatbot for any suitable customer interaction application.

9	1.Account(Acc_no, branch_name,balance) branch(branch_name,branch_city,assets) customer(cust_name,cust_street,cust_city) Depositor(cust_name,acc_no) Loan(loan_no,branch_name,amount) Borrower(cust_name,loan_no) Solve following query: Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc Q1. Find the names of all branches in loan relation. Q2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000. Q3. Find all customers who have a loan from bank. Find their names, loan_no and loan amount. Q4. List all customers in alphabetical order who have loan from Akurdi branch. Q5. Find all customers who have an account or loan or both at bank. Q6. Find all customers who have both account and loan at bank.
10	Account(Acc_no, branch_name,balance) branch(branch_name,branch_city,assets) customer(cust_name,cust_street,cust_city) Depositor(cust_name,acc_no) Loan(loan_no,branch_name,amount) Borrower(cust_name,loan_no) Solve following query: Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc Q1. Find all customers who have both account and loan at bank. Q2. Find all customer who have account but no loan at the bank. Q3. Find average account balance at Akurdi branch. Q4. Find the average account balance at each branch Q5. Find no. of depositors at each branch
11	Account(Acc_no, branch_name,balance) branch(branch_name,branch_city,assets) customer(cust_name,cust_street,cust_city) Depositor(cust_name,acc_no) Loan(loan_no,branch_name,amount) Borrower(cust_name,loan_no) Solve following query: Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc Q1. Find the branches where average account balance > 12000. Q2. Find number of tuples in customer relation. Q3. Calculate total loan amount given by bank. Q4. Delete all loans with loan amount between 1300 and 1500.

	Q5. Delete all tuples at every branch located in Nigdi.
12	solve following join operations. a. Create following Tables cust_mstr(cust_no,fname,lname) add_dets(code_no,add1,add2,state,city,pincode) Retrieve the address of customer Fname as 'Ramesh' and Lname as 'Shinde' b.Create following Tables cust_mstr(custno,fname,lname) acc_fd_cust_dets(codeno,acc_fd_no) fd_dets(fd_sr_no,amt) List the customer holding fixed deposit of amount more than 5000 c. Create following Tables emp_mstr(e_mpno,f_name,l_name,m_name,dept,desg,branch_no) branch_mstr(name,b_no) List the employee details along with branch names to which they belong
13	Create Tables , Account(Acc_no, branch_name,balance) branch(branch_name,branch_city,assets) and solve following Queries. a) Create View on Account table by selecting any two columns and perform insert update delete operations b) Create view on Account and Branch table by selecting any one column from each table perform insert update delete operations c) create updateable view on Account table by selecting any two columns and perform insert update delete operations
14	Consider table Stud(Roll, Att, Status) Write a PL/SQL block for following requirement and handle the exceptions. Roll no. of student will be entered by user. Attendance of roll no. entered by user will be checked in Stud table. If attendance is less than 75% then display the message “Term not granted” and set the status in stud table as “D”. Otherwise display message “Term granted” and set the status in stud table as “ND”
15	

	Write a PL/SQL stored Procedure for following requirements and call the procedure in appropriate PL/SQL block. a. Borrower(Rollin, Name, DateofIssue, NameofBook, Status) b. Fine(Roll_no,Date,Amt) Accept roll_no & name of book from user. Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day. If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.