

C Programming

1. Write a 'C' program to check whether the input number is a perfect square or not.
2. Write a program in 'C' language to sort all the elements in a 4x4 matrix rowwise.
3. Write a 'C' program that finds all the Armstrong numbers between 1 and 1000.
4. Write a function that receives a *float* and an *int* from *main ()*, finds the product of these two and returns the product which is printed through *main ()*.
5. Write a program that will print out all the rotations of a string typed into it. For example for the word "space" they are: *space paces acesp cespa espac*.
6. Implement the following C functions with the help of a program.
 - (a) String length. (b) String concatenation.The program should be implemented without using system-defined functions.
7. Implement a menu-driven C program to find the following
 - (a) Swap two numbers using pointers.
 - (b) Swap two numbers without using a third variable.
8. Write a 'C' program to find the sum of the digits of a given integer.
9. Write a 'C' program to check whether a given string is a palindrome or not.
10. Write a 'C' program to implement the following
 - (a) To check whether a given year is a leap year or not.
 - (b) To find the factorial of a number using recursion.

Database Systems

1. (a) Create the following tables (Use proper referential integrity constraints)

Item (id, name, price)

Customer (id, name, address)

Purchase (item-id, cust_id)

(b) Execute the following queries:

(i) Find out the names of the customers who have purchased any item with price greater than Rs.500/-.

(ii) Find out the name of the most expensive item.

2. (a) Create the following tables (use proper referential integrity constraints)

Employee (id, name)

Department (name, location)

Works (Eid, Did)

(b) Execute the following queries:

(i) Find out the names of the employees working in the department(s) located at 'Chennai'.

(ii) Find out the number of employees who work at 'HR' department.

3. (a) Create the following tables (Use proper referential integrity constraints)

Item (id, name, price)

Customer (id, name, address)

Purchase (item-id, cust_id)

(b) Execute the following queries:

(i) Find out the details of the items purchased by the customers coming from 'Maniktala'.

(ii) Find out the names of the items where prices are greater than the average price of all the items.

4. (a) Create the following tables (use proper referential integrity constraints)

Employee (id, name, salary)

Department (name, location)

Works (Eid, Did)

(b) Execute the following queries:

(i) Find out the details of all the employees who earn most in the 'sales' department.

(ii) Find out the number of employees working in the department(s) located at 'Kolkata'.

5. Create the following relation with at least 4 records.

Employee (id, name, salary, department)

Now execute the following queries:

(a) Find out the names of the employees drawing the highest salary.

(b) Find out the details of the employees working in the sales department.

(c) Increase the salary of each employee by 20%.

6. Create the following relation with at least 4 records.

Product (id, name, price, quantity)

Now execute the following queries:

(a) Find out the name of the most expensive product.

(b) Find out the details of the maximum available product.

(c) Remove everything about the product with id 'P35'.

7. Create the following relation with at least 4 records.

Customer (id, name, address, item purchased)

Now execute the following queries:

- (a) Find out the names of the customers who have purchased gold earrings.
 - (b) Find out the details of the customers from Krishnanagar and Birati.
 - (c) Find out all the different items purchased by all the customers.
8. Create the following relation with at least 4 records.
- Student (roll, name, marks, class)
- Now execute the following queries:
- (a) Find out the number of students in class IX.
 - (b) Find out the name of the student who stood second in class VIII .
 - (c) Display the details of the students with roll numbers 110, 143, 195, 197 and 198.

Data Structures using 'C'

1. (a) Write a program to implement the insertion sort technique.
(b) Write a program to implement the quick sort technique.
2. Write a program to create a binary search tree and traverse in recursive pre-order sequence. Display.
3. Write a menu-driven program to implement the following operations on a singly linked list.
 - (i) Create the list with two nodes.
 - (ii) Insert a new node after identifying a specified node.
 - (iii) Display the list.
4. Write a program to implement the operations of a queue using linked list.
5. Write a program to convert an infix notation to postfix form.
6. Write a program to reverse a singly linked list while traversing once.
7. (a) Write a program to implement the selection sort technique.
(b) Write a program to implement the quick sort technique.
8. Write a program to evaluate a given post-fix arithmetic expression.
9. Write a program that would
 - (i) Create a binary tree with at least 5 nodes;
 - (ii) Traverse it recursively in in-orderly and post-orderly fashion.
10. Write a menu driven program to implement the following operations on a circular linked list.
 - (i) Create the list with at least 5 nodes.
 - (ii) Display the list.

C++

1. Design a class to store a complex number. Now write a program to subtract two complex numbers using a friend function.
2. Design a class to store a height in feet and in inches. Using overloading of '+' operator, add two heights provide by the user.
3. Design a class called 'Arr' which declares an array of integers. From this class, derive another class called 'Sort' which takes inputs to that array and arranges the elements of the array in the descending order.
4. Design a class called 'Arr' which declares an array of integers. From this class, derive another class called 'Srch' which takes inputs to that array and displays the position of an element provided by the user. If the element is not present, display a suitable message.
5. Design a class to store a number in a data member and other necessary member functions to find all the prime factors of that number.
6. Design a class to store the details of an employee. With this class, write a program to find out the employee getting the maximum salary (Consider that there is a single employee with the maximum salary).
7. Write a program to overload the extraction and insertion operator.
8. Write a program to calculate the sum of the following series:
$$x + x^2/2 + x^3/3 + \dots \text{up to } n^{\text{th}} \text{ term, where the values of } x \text{ and } n \text{ are taken as inputs to data members in a class through a suitable constructor.}$$
9. Design a class to store features of different rooms in a hotel. Use suitable member functions to find the details of the costliest room.
10. Design a class to store the distance in meters and centimetres. By overloading of '>' operator, find the greater of two distances provided as inputs.
11. Design a class called Employee with the general attributes of an employee in an organization. From this class, derive two classes called 'Supervisor' and 'Subordinate' to store special features of a supervisor and a subordinate respectively. Perform input-output operations on these derived classes.
12. Design a class called 'Power' with at least two data members-base and exponent. With a constructor, initialize these members and with suitable member function, find $base^{\text{exponent}}$.
13. Write a class called 'Str' to store a string. Write suitable member functions to compare the equality of two strings given as inputs.
14. Use a friend function to multiply a vector stored in a class by a scalar quantity, given as input.