EXTRA LAB EXERCISES FOR IMPROVING PROGRAMMING LOGIC

Q1:- Write a C program that acts as a simple calculator. The program should take two numbers and an operator as input from the user and perform the respective operation (addition, subtraction, multiplication, division, or modulus) using operators.

• Challenge: Extend the program to handle invalid operator inputs.

D:\tops_assigenment\lab_prectic\simple_calculater.c

Q2:- Write a C program that takes an integer from the user and checks the following using different operators:

Whether the number is even or odd.

Whether the number is positive, negative, or zero.

Whether the number is a multiple of both 3 and 5.

D:\tops_assigenment\lab_prectic\chack_condition.c

Q3:- Write a C program that takes the marks of a student as input and displays the corresponding grade based on the following conditions:

o Marks > 90: Grade A o Marks > 75 and <= 90: Grade B

o Marks > 50 and <= 75: Grade C

o Marks <= 50: Grade D

• Use if-else or switch statements for the decision-making process.

D:\tops_assigenment\lab_prectic\result.c

Q4:- Write a C program that takes three numbers from the user and determines:

o The largest number.

o The smallest number.

D:\tops_assigenment\lab_prectic\function.c

Q5:- Write a C program that checks whether a given number is a prime number or not using a for loop.

• Challenge: Modify the program to print all prime numbers between 1 and a given number.

D:\tops_assigenment\lab_prectic\q5_prime.c

Q6:- Write a C program that takes an integer input from the user and prints its multiplication table using a for loop.

• Challenge: Allow the user to input the range of the multiplication table (e.g., from 1 to N).

D:\tops_assigenment\lab_prectic\multiplication.c

Q7:- Write a C program that takes an integer from the user and calculates the sum of its digits using a while loop.

• Challenge: Extend the program to reverse the digits of the number.

D:\tops_assigenment\lab_prectic\rev1.c

Q8:- Write a C program that accepts 10 integers from the user and stores them in an array. The program should then find and print the maximum and minimum values in the array. • Challenge: Extend the program to sort the array in ascending order.

D:\tops_assigenment\lab_prectic\minimum_array.c

D:\tops_assigenment\lab_prectic\max_array.c

Q9:- Write a C program that accepts two 2x2 matrices from the user and adds them. Display the resultant matrix. • Challenge: Extend the program to work with 3x3 matrices and matrix multiplication.

D:\tops_assigenment\lab_prectic\2d_3d_array.c

Q10:- Write a C program that takes N numbers from the user and stores them in an array. The program should then calculate and display the sum of all array elements. • Challenge: Modify the program to also find the average of the numbers.

D:\tops_assigenment\lab_prectic\sum-everg_array.c

Q11:- Write a C program that generates the Fibonacci sequence up to N terms using a recursive function.

• Challenge: Modify the program to calculate the Nth Fibonacci number using both iterative and recursive methods. Compare their efficiency.

D:\tops_assigenment\lab_prectic\recourcive_function.c

Q12:- Write a C program that calculates the factorial of a given number using a function. • Challenge: Implement both an iterative and a recursive version of the factorial function and compare their performance for large numbers.

D:\tops_assigenment\lab_prectic\q12_find_fact.c

Q13:- Write a C program that takes a number as input and checks whether it is a palindrome using a function. • Challenge: Modify the program to check if a given string is a palindrome.

D:\tops_assigenment\prectice\revers.c

Q14:- Write a C program that takes a string as input and reverses it using a function. • Challenge: Write the program without using built-in string handling functions.

D:\tops_assigenment\lab_prectic\q14_revers_string.c

Q15:- Write a C program that takes a string from the user and counts the number of vowels and consonants in the string. • Challenge: Extend the program to also count digits and special characters.

D:\tops_assigenment\lab_prectic\q15_count_string.c

Q16:- Write a C program that counts the number of words in a sentence entered by the user. • Challenge: Modify the program to find the longest word in the sentence.

D:\tops_assigenment\lab_prectic\q16_word_count.c

Extra Logic Building Challenges

Q:-1 Write a C program that checks whether a given number is an Armstrong number or not (e.g., 153 = 1^3 + 5^3 + 3^3). • Challenge: Write a program to find all Armstrong numbers between 1 and 1000.

D:\tops_assigenment\lab_prectic\q1_armstrong_num.c

D:\tops_assigenment\lab_prectic\q1_armstrong_num2.c

Q:-2 Write a C program that generates Pascal's Triangle up to N rows using loops. • Challenge: Implement the same program using a recursive function.

D:\tops_assigenment\lab_prectic\q2_Pascal's_Triangle.c

Q:-3 Write a C program that implements a simple number guessing game. The program should generate a random number between 1 and 100, and the user should guess the number within a limited number of attempts. • Challenge: Provide hints to the user if the guessed number is too high or too low.

D:\tops_assigenment\lab_prectic\number_gguessing_game.c