Introduction to C Programming

IT 1st Semester, Spring 2022

Lab Programs List

Lab1 (Basics)

1. Write a program to print your name, ID, and JEE rank/percentile using printf statements ONLY.
2. Write a program to find the addition, subtraction, multiplication and division of two FLOAT numbers.
3. Write a program to take salary as input and calculate tax, PF deduction, expenditure and net savings.
4. Write a program to explore various functions of math.h header file – pow, sqrt, sin, cos, tan.
5. Swap two numbers using a third variable.
6. Swap two numbers without using a third variable.

Lab2 (Conditional flow – if-else and switch case)

1. Find the largest of 5 numbers.
2. Find if a number is even or odd.
3. Find if a number is positive, negative, or zero.
4. Find the type of triangle from its sides – scalene, isosceles, right-angled, isosceles right-angled, or equilateral.
5. Perform basic arithmetic using switch case – take user choice of operation (addition, subtraction, multiplication, division) and print outputs accordingly.
6. Print the number of days of a month based on the month number, using switch case.

Lab3 (Looping – while, for, do-while)

1. WAP to print all even numbers using one loop and odd numbers using another loop (both in the same program). Take max number (limit) as input.
2. WAP to check if the number entered by a user is even or odd. Continue checking until user wishes to exit; keep looping between user choice and entered number.
3. WAP to print the factorial of a number using do-while loop. Be careful about zero.
4. WAP to print all prime numbers in a given range n1~n2.
5. WAP to find the power of a given number without using the pow( ) function. Use loops. If power is negative, print as a fraction instead of decimals.
6. WAP to find if a 3-digit number is an Armstrong number or not. A 3-digit Armstrong number is one that is equal to the sum of cubes of its individual digits.
7. WAP to use loops and if-else to print Pascal’s triangle for n rows. Do not hard-code the triangle into printf statements.

Lab4 (Arrays)

1. Write a program to perform the following operations on an array A[10]: insert an element in between, delete an element in between, print the array, reverse the array (into new array).
2. Write a program to take an array A[10] with values from user input, and find the following: largest element, smallest element, mean value, even-valued elements, and odd-valued elements.
3. Write a program to perform the following: addition of two 2x3 matrices, multiplication of two compatible matrices, transpose of a 3x4 matrix, and determinant of a 3x3 matrix.
4. Write a program to search and print the first occurrence of an element in an array. Print “not found” if absent. Take 10 numbers as the array elements input and an 11th number as the search element.
5. Write a program to print the most frequent element in a 15-number array. If all are unique, print accordingly.

Lab5 (Functions)

1. Build function(s) to find the HCF and LCM of two numbers.
2. Build function(s) for basic temperature conversion – take user input in one unit (Celsius, say) and convert to the other two units (Fahrenheit and Kelvin, if Celsius is input).
3. Build function(s) to compute the roots of a Quadratic equation of the form “$ax^2 + bx + c = 0$”.
4. Build function(s) to find the prime factors of a number.
5. Build function(s) to reverse a number and check if it is a palindrome. Be careful with negative numbers.
6. Find the factorial of a number using two separate functions in a single program:
   a. Function that loops simply
   b. Function that uses recursion
7. Build function(s) to perform linear search

Lab6 (Pointers Part1)

1. WAP to find the sum of array elements using pointers.
2. WAP to check the equality of two same-sized arrays using pointers.
3. WAP to swap two same-sized arrays using pointers.
4. WAP to reverse an array using pointers.
5. WAP to perform linear search on an array using pointers.
6. WAP to use pointers for denoting a 2D array (matrix) and find the following:
   a. Mean value
   b. Max and Min values
   c. Median value (middle of all)
   d. Mode value (most frequent of all)
7. WAP to perform matrix arithmetic using pointers. Input two NxN sized matrices and find sum, difference and product.
8. WAP to check the following properties for a 3x3 matrix:
   a. Symmetric or asymmetric or skew-symmetric
   b. Identity matrix
   c. Invertible matrix

Lab7 (Pointers Part2 – Strings)

1. Use pointers and count the following in a given string:
    a. No. of characters
    b. No. of words
    c. No. of uppercase letters
    d. No. of digits
    e. No. of symbols (excluding spaces)
2. Use pointers and delete the first occurrence of a single character in a given string.
3. Use pointers and delete all white spaces in a given string.
4. Use pointers and find the first occurrence of a given substring in a given string. Return "Not found" otherwise.
5. Use pointers and reverse a string (manipulate a duplicate of the original string). Also, check if it is a palindrome.
6. Use pointers and define your own functions that do the following operations:
    a. String comparison
    b. String concatenation
    c. String copying

Lab8 (Structures)

1. Use Structures and find and print the difference between two time periods (hh:mm:ss format).
2. Modify question1 to find and print the difference between two dates (dd:mm:yyyy format).
3. Find a person's age from their birth date.
4. Perform basic arithmetic operations – addition, subtraction, multiplication, and division – on two complex numbers.
5. Write the truth tables of various logic gates using structures.
6. Use structures to initialize various triangles (by sides and/or angles) and print their types.
7. Modify question6 to initialize using vertex coordinates instead of sides/angles.

Lab9 (Miscellaneous)

1. Sort a list of strings in dictionary/alphabetical/lexicographical order.
2. Print an input number in words. E.g. 42 = forty two
3. Delete a substring from a given string. E.g. deleting "as" from "assassination" gives "ssination"
4. Insert a substring after every occurrence of a character in a string. E.g. inserting "as" after every "e" in "evermore" gives "easveasrmoreas".
5. Print the distance travelled at each second, using given value of acceleration and computing distance and velocity repeatedly at each time instant.
    a. Format: "At time t=<value>, u=<value>, d=<value>"
    b. E.g. "At time t=10, u=10, d=500"
6. Generate a pyramid of * (asterisk) symbols.
7. Generate a diamond of @ (at) symbols.

8. Find the divisibility of a number by 2,3,4,5,6,8,9,10,11 using divisibility rules.
9. Iteratively find and print the topper student in a student list as long as the user is giving inputs. Use dynamic memory allocation.
10. Find the mean, median, and mode of a randomly generated array of integers. Use pointers.

Lab10 (File Handling)

1. WAP to read data from the keyboard, write into a file called “INPUT.txt”, again read from the file, and print the data on the screen.
2. A file named “DATA.txt” contains a series of integer numbers. WAP to read these numbers and write odd numbers into a file “ODD.txt” and even numbers into “EVEN.txt” and then display the contents of both files.
3. Open a file called “Student.txt” and write data into it in the format: ID, name, branch, grade. Read the file and display the contents in tabular manner.
4. Copy the contents of one file to another.
5. Compare two files and return 1 if equal and 0 otherwise.
6. Modify question3 to create a BILL.txt file to store: ItemID, Name, Quantity, Rate, Price. Read the prices and print the items as a bill with the total bill amount.