



**PROBLEM-SOLVING USING COMPUTERS LAB
(CS1031)**

LAB MANUAL

I/II Semester, B. Tech.

(2023-24)

Faculty of Engineering

CONTENTS

- 1. Instructions to Students**
- 2. Lab Evaluation Scheme**
- 3. List of Experiments**
- 4. References**
- 5. Additional Exercises**

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab practice.
2. In each lab session, at least 4 programs should be executed and shown to the lab instructor. Remaining programs should be executed and shown as home assignments.
3. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
4. Students should bring and maintain an observation book exclusively for the lab.
5. Once the experiment(s) is/are executed, they should show the program and results to the instructors and copy the same in their observation book.
6. Students are also required to draw the corresponding flowcharts in the observation book during the lab hours itself which will be evaluated later. Drawing of flowcharts are optional for the programs from week 6. Prescribed textbooks and class notes can be kept ready for reference if required.
7. Students are supposed to implement the given experiment individually.
8. While conducting the experiments students should see that their programs would meet the following criteria:
 - a) Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
 - b) Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
 - c) Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs, and outputs.
 - d) Statements within the program should be properly indented.
 - e) Use meaningful names for variables and functions. Do not use any random names.
 - f) Make use of Constants and type definitions wherever needed.
9. Questions for lab tests and exams need not necessarily be limited to the questions in the manual but could involve some variations and/or combinations of the questions.
10. Practice of additional exercises would be considered as formative assessment and may not be written in lab records.

** Above-mentioned instructions can be modified based on the context of the lab.*

LAB EVALUATION SCHEME

Total marks: 100

1. Continuous evaluation = 60 marks

This consists of 3 evaluations of 20 marks each throughout the semester.
Each evaluation would consist of the following:

- Observation of lab records: 5 marks
- Write-up and Program Execution: 5 marks
- Lab quiz: 10 marks

2. External Exam = 40 marks

- Write-up and execution: 15 marks
- Viva: 10 marks
- Quiz: 15 marks

LIST OF EXPERIMENTS

Lab 1. Algorithms and Flow Charts

1. Algorithm
 - a) Introduction
 - b) Different features of an algorithm
 - c) Sequence
 - d) Decision
 - e) Repetition
 - f) More on algorithms (examples or use cases)
2. Flowchart drawing using modern tool
 - a) Introduction and significance
 - b) Different shapes
 - c) Examples

Lab 2. Working with Linux Commands

1. Introduction to the Linux operating system and its commands (e.g. *cd*, *mv*, *mkdir*, *cp*, *rm*, *cat*, *ls*, *mv*, *pwd*, etc).

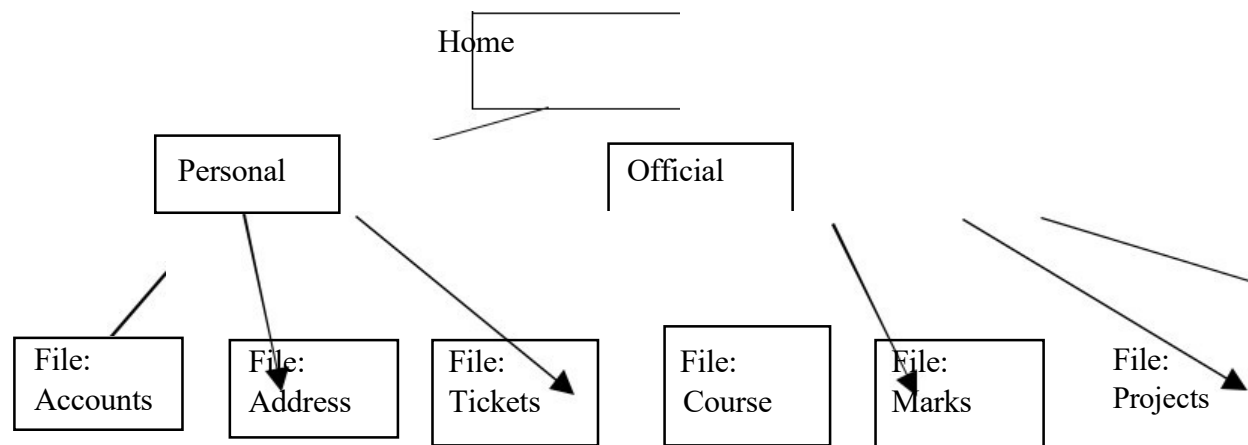


Fig. 1. Sample Directory Structure

Refer the directory structure shown in **Fig. 1**. Accomplish the following tasks, in sequence.

- a. Create the directory structure as in Fig. 1 in your home directory.
 - b. Rename each file in **Personal** with the prefix P. i.e., the file **Accounts** in **Personal** directory should be renamed as **Paccounts**.
 - c. Move the file named **Paddress** to a newly created directory in **Home** from its current location.
 - d. Create a directory **Backup** under **Home** directory and move all the contents of **Personal** into **Backup**.
 - e. Remove all the contents of **Personal** directory.
 - f. Now rename the **Backup** directory as **Personal**.
2. Understanding the GEDIT editor and execution of source code.

Lab 3. Formula based C Programs

1. Write a program to convert the time in seconds to hours, minutes, and seconds. (1 hr =3600 sec).
2. Write a program to find the sum of the digits of a four-digit number (ex. 1234 sum=10) (without using a loop).
3. Write a program to convert the temperature given in Fahrenheit to Centigrade and Centigrade to Fahrenheit. Hint: $C=5/9(F-32)$.
4. Write a program for converting distance in mm to cm, inch, feet (1 cm =10mm, 1inch=2.5cm, 1 feet =12 inches).
5. Write a program to find out the distance between two points e.g. (x1, y1) and (x2, y2).
Hint: Distance= $\sqrt{(x2-x1)^2 + (y2-y1)^2}$
6. Write a program to evaluate the area of the circle Area = Pi * R²
7. Write a program to interchange the values of two variables using a third variable.
8. Write a program to interchange the values of two variables without using a third variable.

Lab 4. Control Structures: If statement

1. Write a program to check whether the given number is odd or even.
2. Write a program to check whether a given year is a leap year or not.
3. Write a program to find the roots of a quadratic equation.
 - a. Hint: root = -b +/- sqrt(b²-4ac)/2a
4. Write a program to find the total no. of days for a given number of months counting from January month.
 - a. Example: m=3, days=31+ (28 Or 29) + 31.
5. Write a program to take two numbers as an input and find whether one number is a multiple of the other or not.
6. Write a program that returns a letter grade based on a quiz score. The input will be the integer score from a ten-point quiz.
 - a. The letter grades are assigned by:
 - b. 9-10“A”7-8“B”5-6“C”3-4“D”<3“F”
7. Write a program that takes three sides of a triangle input and calculates its area, if these conditions are satisfied a+b>c, b+c>a, a+c>b, calculate area=(a+b+c)/2
8. Write a program to check whether the given character is a vowel, consonant, or digit.

Lab 5. Control Structures: Switch

1. Write a program to program to calculate an area of a circle, a rectangle, or a triangle depending on the user's choice.
2. Write a program to remove all the break statements from Ex-1 (with switch-case construct) and try to execute the program with a few inputs. Observe the difference.
3. Write a program to program to input the number of week's days (1-7) and translate to its equivalent name of the day of the week (e.g., 1 to Sunday, 2 to Monday)
4. Write a program to design a calculator that performs addition, subtraction, minus, and division operation. This program inputs two operands and an operator and then displays the calculated results.
5. Write a program to calculate a bill for internet browsing. The conditions are given below: Minimum Rs. 200 for up to 100 calls.
Plus, Rs. 0.60 per call for the next 50 calls.
Plus, Rs. 0.50 per call for the next 50 calls.

Plus, Rs. 0.40 per call for any call beyond 200 calls

6. Write a program to calculate an amount of a telephone bill for the following criteria. (Without Loop)

Calls	charge per call (Rs.)
1-150	0
151-250	.9
251-400	1.2
401 onwards	1.5

7. Write a program to calculate the amount of an electricity bill for the following criteria. (Without Loop)

Units	charge per unit (Rs.)
First, 1-100 up to	0
Next, 101-200 up to	1.5
Next, 201-400 up to	2.5
401 onwards	3.5

8. Write a program to calculate the discount in rupees for the following criterion. (Without Loop)

Cost price	discount
≥ 800	25%
500-800	20%
< 500	no discount (0%)

Lab 6. Control Structures: Loops

1. Write a program to take N as input and print the odd numbers in descending order.
2. Write a program to print the Fibonacci number.

Hint: (Fibonacci series is 0, 1, 1, 2, 3, 5, 8,)

3. Write a program to find whether the given number is prime or not.
4. Write a program to convert the decimal number into binary to

decimal. Ex: $1101 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13$

5. Write a program to reverse a given number
Ex: $1234 \text{ reverse} = 4 * 10^3 + 3 * 10^2 + 2 * 10^1 + 1 * 10^0 = 4321$
6. Write a program to find the sum of n terms of the sin series $\sin(x) = x - x^3 + x^5 - x^7$
7. Write a program to check whether a given integer no. is palindrome or not.
8. Write a program to check whether the given number is Armstrong or not. An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^3 + 7^3 + 1^3 = 371$.

Lab 7. Control Structures: Nested Loops

1. Write a program to print different patterns using nested loops. 1

```
2      3
4      5      6
7      8      9      10
11     12     13     14     15
```

```
1
2      2
3      3      3
4      4      4      4
5      5      5      5      5      5
```

```
5      5      5      5      5
4      4      4      4
3      3      3
2      2
```

1

```
1
2      4
3      5      7
6      8      10     12
9      11     13     15     17
```

5. Write a program to generate the multiplication table for n numbers up to k terms (nested loops).

Hint:

```
1  2  3  4  5....K2
4  6  8 10 ... 2*k
.....
.....
n..... n*K
```

6. Write a program to print the Fibonacci numbers that fall in the given range.
7. Write a program to print that Nth the prime number.
8. Write a program to print that Nth the Armstrong number.

Lab 8. 1-D Array

1. Write a program to find the largest and smallest element in an array.
2. Write a program to find the sum of odd index numbers in an array.
3. Write a program to print the subarray that lies between the two indexes.
4. Write a program to find the number of positive numbers, negative numbers, odd numbers, even numbers, and the number of 0 of an array.
5. Write a program to reverse an array with an auxiliary array.
6. Write a program to check whether an array is sorted or not.

7. Write a program to arrange the elements of an array in ascending order by a simple sorting method. (Selection sort/bubble sort)
8. Write a program to take an array of 10 elements. Split it into the middle and store the elements in two different arrays. E.g.- Initial array:

58	24	13	15	63	9	8	81	1	78
----	----	----	----	----	---	---	----	---	----

After splitting:

58	24	13	15	63
----	----	----	----	----

9	8	81	1	78
---	---	----	---	----

Lab 9. 2-D Arrays

1. Write a program to take 3 student marks in 5 subjects. Print the total marks of each student and the average marks of each subject.
2. Write a program for searching for an element in the matrix and counting the number of occurrences of that element.
3. Write a program to multiply two matrices.
4. Write a program to check if the given matrix is a magic square or not.
5. Write a program to find whether a given matrix is symmetric or not.

Hint: $A = A^T$

6. Write a program to find the trace and norm of a given square matrix.

Hint: Trace = sum of principal diagonal elements

Norm = sort (sum of squares of the individual elements of an array)

Lab 10. Strings

1. Write a program to change all lower-case letters into upper case in a sentence.
2. Write a program to find the last occurrence of a particular character.
3. Write a program to concatenate/length/copy two strings using the library function.
4. Write a program to count the number of words in a sentence.
5. Write a program to reverse a string.
6. Write a program to find the string length of a string without using the predefined function.
7. Write a program to find the substring of a given string.
8. Write a program to check if the given string is a palindrome or not.

Lab 11. Functions

1. Write a program to find the factorial of a number using a function.
(Ex: $5! = 5*4*3*2*1$. Use a function Fact to evaluate factorial & print the result).
2. Write a program to find the maximum of a given set of numbers using functions. (Use a function Max and return the result to the main function)
3. Write a program to find the GCD of two numbers recursively. (Ex: GCD of 9, 24 is 3)

4. Write a program to check whether the given number is prime or not. Using this function generates first n prime numbers using the above function.
5. Write a function to generate the nth Fibonacci term using recursion. Print first N Fibonacci terms using this function.
Hint: (Fibonacci series is 0, 1, 1, 2, 3, 5, 8,)
6. Write a program to check if the given string is a palindrome or not, using the string handling function.
7. Write a function **Sort** for sorting a list of names which will use a function **compare** to compare two names. (Selection /bubble Sort may be used).

Lab 12. Pointers

1. Write a program to access two integers using pointers and add them.
2. Write a program to find out the greatest and the smallest among the three numbers using pointers.
3. Write a program to determine the length of a character string using a pointer.
4. Write a program to compute the sum of all elements stored in an array using a pointer.
5. Write a program to determine whether a substring (string 1) is in the main string or not. If present, return the pointer of the first occurrence.

Lab 13. Structures

1. Write a program to define a structure personal that would contain the person's name, date of joining, and salary. Using this structure write a program to read this information for one person from the keyboard and print the same on the screen.
2. Write a program to create an array of student structures to store the roll no., name, and marks in 3 subjects. Input the details of N students into the array and display roll no., name, and total marks of each student in decreasing order of total marks.
3. Write a program to create an array of employee structures to store emp-no, name, basic salary, and HRA. Input the details of N employees and display emp-no, name, basic, HRA, and net salary. Display the details of all employees whose net salary is more than the average net salary of all employees.
4. Write a program to create a structure named Date having day, month, and year as its elements. Store the current date in the structure. Now add 45 days to the current date and display the final date.

Lab 14. End-term Exam

REFERENCES

1. E. Balagurusamy, “*Programming in ANSI C*”, 7th Edition, McGraw Hill Publication, 2016.
2. Y. P. Kanetkar, “*Let us C*”, 12th Edition, BPB Publication, 2014.
3. B. W. Kernighan, D. M. Ritchie, “*The C Programing Language*”, 2nd Edition, Prentice Hall of India, 2014.
4. Gottfried, “*Schaum's Outline Series: Programming with C*”, 3rd Edition, McGraw Hill Publication, 2012.

ADDITIONAL EXERCISES

1. Write a program to find the determinant of a square matrix.
2. Write a program to check whether a saddle point exists in each matrix. (i.e., an element which is minimum in a row and maximum in the column)
3. Write a program, which generates all the possible combinations of a 3-digit number.
4. Given a list of N numbers in the range 1 to N (not necessarily in the order) and out of that one number is missing (i.e., only N-1 inputs), write a program to find the missing number.
5. Two sets are represented by arrays A and B. Find AUB, A^B, and A-B.
6. Given 4 numbers 1, 2, 3, and 4. Find all possible 4-digit numbers that can be formed using these 4-digits. From these numbers, generate the permutation of the 4 digits (i.e., 4 digits are distinct)
7. Write a program to find the median of N numbers.
8. Write a program to check whether the given number is Armstrong Number.
9. Write a program that examines all the numbers from 1 to N, displaying all those for which the sum of the cubes of all the digits equals the number itself.
10. Input N numbers and place only those numbers whose Pth position (specified by user) is divisible by 3 into an array. (E.g.: The numbers input: 1223, 1243, 1232; position P=4 (left to right) then the resultant array will {1223,1243})
11. A number has all its digits squared and added together to make a new number. This process is repeated until a 1 is obtained, then the original number is described as "HAPPY". If a 1 is never obtained, then the original number is said to be "SAD".
{e.g.: $19 = 1^2 + 9^2 = 82 \Rightarrow 8^2 + 2^2 = 68 \Rightarrow 6^2 + 8^2 = 100 \Rightarrow 1^2 = 1$ }
12. Write a program to delete blanks from a given string.
13. Write a program to display following output for N number of lines.

```

                1
              1 2 1
            1 2 3 2 1
          1 2 3 4 3 2 1
        1 2 3 2 1
      1 2 1
    1
  
```

14. Find the inverse of a matrix A.
15. Find the solution to the system of simultaneous equations using Gauss- Jordan elimination method.
16. Display the list of perfect squares in the given mXn matrix.
17. Given a string S1=" This is a way to do this" and a string S2="is" find the positions of occurrence of S2 in S1.
18. Replace odd numbers in a matrix by zeroes.
19. Display all the Pythagorean triplets up to n. (Eg: $3^2 + 4^2 = 5^2$).
20. Write a program to find the transpose of a matrix without using another matrix.
21. WAP to print the abbreviation of a given name. (Eg: Central Computing Facility => C.C.F)

22. WAP to simulate an arithmetic calculator.
23. Replace each element of the matrix by its factorial.
24. Find the odd numbers in the matrix and display its row and column number.
25. Convert 3×3 matrix to a 4×4 matrix.

Eg: Original Matrix:

1	2	3
1	2	3
1	2	3

New Matrix:

1	2	3	6
1	2	3	6
1	2	3	6
3	6	9	18

26. WAP to print all combinations of a given word. (Eg: Rose => oser, osre.... Etc.)
27. Write an n digit number in words. (236=two three six)
28. Find the volume of a cylinder and a cube.
29. Find the LCM of two numbers.
30. Find all the primes in the multiplication tables ranging from 2 to 5 and store them in the array without having duplicates.
31. Check all the rows, columns, principal diagonal, and secondary diagonal for palindromes in an integer matrix of any order.
32. WAP to find the digital roots of numbers for multiplication tables ranging from 2 to 5 and store those forty resultant numbers in a matrix of order 4X10. (Digital root of number is a single digit derived from adding all the individual digits of the number repeatedly. (Eg: digital root of 2456= 2+4+5+6=17, again 1+7=8---> digital root)
33. Generate

				1					
				1		1			
			1		2		1		
		1		3		3		1	
	1		4		6		4		1

34. Write a program to read a line, encode the line and display the original and encoded form.

The encode should be:

a b c d z
z y x w a

35. Write a program to find the time of the day when
 - a Hour and Minute hand are exactly opposite to each other.
 - b Hour and Minute hand are overlapping each other.