

PROBLEM SOLVING TECHNIQUES USING C – LAB

COURSE CREDITS: 02

NO. OF HOURS: 60

COURSE OUTCOMES (COS):

CO1 : understand and apply basic C programming concepts such as variables, conditionals, and loops.

CO2: Comprehend the logic behind conditional statements and loops in C programming.

CO3: Apply techniques to solve problems involving arrays, strings, and pointers.

CO4: Analyse and break down problems to implement recursive functions and simulate complex operations using C programming constructs.

CO5: Develop and write programs to perform tasks like file operations, matrix manipulations, and demonstrating advanced concepts such as structures, unions, and different methods of parameter passing.

PART A

1. To read the radius of a circle and to find area and circumference.
2. To read three numbers and find the biggest of three using a nested if statement.
3. To read the numbers from the keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
4. To design the following pattern using nested for loop:

```
*
* *
* * *
* * * *
* * * * *
```

a.

5. To calculate the sum of all elements in an array of n integers.
6. To read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
7. To generate Fibonacci series using a recursive function.
8. To demonstrate pointer declaration, initialization, and accessing the value of a variable using a pointer.
9. Write a C program to add two numbers using command line arguments.

PART B

1. To read a number, find the sum of the digits, reverse the number and check it for palindrome.
2. To read the percentage of marks and to display appropriate messages. If a percentage is 70 and above- Distinction, 60-69 – First Class, 50-59 – Second Class, 40-49 Pass, below 40 – Fail. (Demonstrate of if-else ladder)

3. To simulate a simple calculator with addition, subtraction, multiplication, division and it should display the error message for division of zero using switch case.
4. To read marks scored by n students and find the average of marks (Demonstration of single dimensional array).
5. To remove Duplicate Element in a single dimensional Array.
6. To read, display and add two m x n matrices using functions.
7. Write a C program to show the difference between 'call by value' and 'call by reference' by swapping two numbers.
8. To demonstrate the difference between structure & union.
9. Write a C program to copy contents from one text file to another text file.