.....
.....

Problem Solving Using C Lab

(BMC-102)

LAB MANUAL

COURSE: MCA

SEM: 1st

Department of Master of Computer Applications

G. L. BAJAJ INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Plot no. 2, Knowledge Park III, Gr. Noida

.....
.....

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Vision, Mission & PEOs of the Department

VISION

To become a center of repute and developing the computer professionals who can respond to social and global challenges

MISSION

- To impart quality education with sound practical knowledge for societal and global recognition
- To provide exposure and awareness about Industry needs and challenges through mutual association
- To provide environment for innovation, incubation and entrepreneurship
- To develop moral values and ethics in our graduates

Program Educational Objectives (PEOs)

PEO 1: To progress their career in Industry, Academia, Research, **e**ntrepreneurial pursuit, consultancy firms and other technological enabled services.

PEO 2: To excel in career as an individual or in a team; by adopting ethics and professionalism and communicate seamlessly with cross culture and interdisciplinary teams.

PEO 3: To continue a lifelong learner in computing and contributes in societal growth.

.....
.....

Program Outcomes/Program Specific Outcomes

Program Outcomes(POs)

1. Computational Knowledge: Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

2. Problem Analysis: Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.

3. Design /Development of Solutions: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

4. Conduct investigations of complex Computing problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

6. Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.

7. Life-long Learning: Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

8. Project management and finance: Demonstrate knowledge and understanding of t h e c o m p u t i n g and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

……
……

9. Communication Efficacy: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

10. Societal and Environmental Concern: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices.

11. Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

12. Innovation and Entrepreneurship Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Program Specific Outcomes

Program Specific Outcomes (PSO's)

PSO 1: Acquire knowledge and Apply problem solving strategies in cutting edge technologies

PSO 2: Design and Development of software product and services to social and global requirements.

.....
.....

Cos	COURSE OUTCOMES	BLOOM'S KNOWLEDGE LEVEL (KL)
CO1	Write, compile, debug and execute program in a C programming environment.	K3
CO2	Write programs that incorporate use of variables, operators and expression along with data types.	K3
CO3	Write programs for solving problems involving use of decision control structure and loops.	K3
CO4	Write programs that involve the use of arrays, structures and user define functions.	K3
CO5	Write program using graphic and file handling operations.	K3

Mapping of Program Outcomes with Course Outcomes(COs)

CO-PO Matrix												
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	-	-	-	-	-	-	-	-	-	-	-	-
CO2	-	-	-	-	-	-	-	-	-	-	-	-
CO3	-	-	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	-	-	-	-	-	-	-	-
CO5	-	-	-	-	-	-	-	-	-	-	-	-

CO-PSO Matrix		
CO	PSO1	PSO2
CO1	1	1
CO2	1	1

.....
.....

CO3	2	2
CO4	2	2
CO5	2	2

List of Experiments

CATEGORY	PROGRAM NO	TITLE	CO
BASIC INPUT OUTPUT OPERATIONS	1.	Write a program to input and add two numbers	**co1**
	2.	Write a program to convert Fahrenheit to Celsius and Celsius to Fahrenheit.	**co1**
	3.	Write a program to calculate Area of a rectangle	**co1**
	4.	Write a program to Calculate Area and Perimeter of a Circle	**co1**
	5.	Write a program to determine the roots of quadratic equation	**co1**
	6.	Write a program to Calculate Simple Interest	**co1**
	7.	Write a program to Calculate Compound Interest	**co1**
	8.	Write a program to swap two numbers using third variable, without using third variable, within single statement using comma operator.	**co2**
	9.	Write a program to input two numbers and find the greatest number using ternary operator.	**co2**

.....
.....

	10.	Write a program to find odd/even number using ternary operator.	**CO2**
	11.	Write a program to Input three numbers and find greatest number using ternary operator.	**CO2**
	12.	Enter arithmetic operator from user, perform operation on two numbers according to the operator using ternary operator.	**CO2**
	13.	Find odd/even number using bitwise operator.	**CO2**
	14.	Write a program to swap two numbers using bitwise operator.	**CO2**
	15.	Find exact power of 2 of a given number using bitwise operator.	**CO2**
CONDITIONAL & SWITCH CASE STATEMENTS	**1**	Write a program to input the annual taxable income and compute the tax according to the given condition.	**CO3**

Total Annual Taxable Income	Tax Rate
Up to Rs. 1,00,000	No Tax
From 1,00,001 to 1,50,000	10% of the income tax exceeding Rs. 1,00,000
From 1,50,001 to 2,50,000	Rs. 5000 + 20% of the income tax exceeding Rs. 1,50,000
Above 2,50,000	Rs. 25000 + 30% of the income tax exceeding Rs. 2,50,000

……
……

	2	Write a program to calculate the monthly telephone bills as per the following rule. • Minimum Rs. 100 for up to 50 calls • Plus Rs. 0.80 per call for next 50 calls • Plus Rs. 0.60 per call for next 100 calls • Plus Rs. 0.40 per call for any call beyond 200 calls	**CO3**
	3	Write a currency program that tells you how many 500,200,100,50,20,10,5,2 and 1 Rs. notes will be needed for a given amount of money.	**CO3**
	4,	Write a program that estimated the price of rings for an online shop that sells rings with custom engravings. The online shop has the following price structure: Gold plated rings have a base cost of Rs. 500, and you charge 70.30 per engraved unit. Solid gold rings have a base cost of Rs. 1000, and you charge 10.40 per engraved unit. Silver plated rings have a base cost of Rs. 300, and you charge 5.10 per engraved unit. Find the total cost as per the ring purchase by customer.	**CO3**
LOOP CONSTRUCT	1	Write a C program to print the of the Fibonacci series	**CO3**
	2	Write a program, enter a number from user check that number is prime or not.	**CO3**
	3	Write a program to print a multiplication table of n number in reverse order.	**CO3**
	4	Write a program, enter a number from user reverse that number.	**CO3**
	5	Write a program, enter a number from user check that is number is Armstrong or not.	**CO3**
	6	Write a program to find out the value of x raised to the power y, where x and y are positive integers	**CO3**
	7	Write a C program to find the H.C.F. of two numbers.	**CO3**

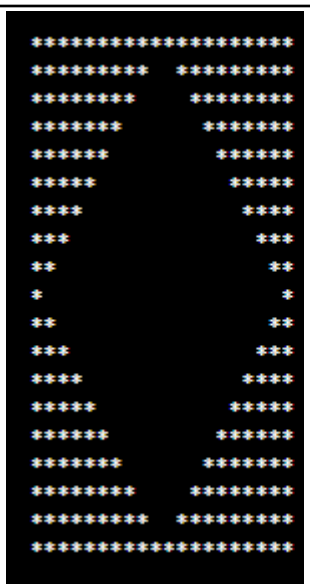
……
……

	8	Write a program to receive a five-digit no and display as like 24689: 2 4 6 8 9	**co3**
	9	**Number Guessing Game** We will write a program that generates a random number and asks the player to guess it. If the player's guess is higher than the actual number, the program displays "Lower number please". Similarly, if the user's guess is too low, the program prints "Higher number please". When the user guesses the correct number, the program displays the number of guesses the player used to arrive at the number. Hint: Use loop & use a random number generator.	**co3**
	10	Write a program in C language to find harmonic series and its sum up to n. Enter the range: 10 Harmonic Sequence is: 1/1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 + 1/9 + 1/10 sum of harmonic series: 2.928968	**co3**
	11	Write a program in C to print arithmetic progression and its sum. Input: First term: 2, Common difference: 4 Number of terms: 10 Output: Enter the first term of the A.P.: 2 Enter the common difference: 4 Input number of terms in the series: 10 Arithmetic Progression : 2 6 10 14 18 22 26 30 34 38 The sum of the AP series is: 200	**co3**
	12	**Square Root**	**co3**

.....
.....

		Given a non-negative integer x, return *the square root of* x *rounded down to the nearest integer*. The returned integer should be **non-negative** as well. You **must not use** any built-in exponent function For example: do not use pow(x, 0.5) **Example 1: Input:** x = 4, **Output:** 2 **Explanation:** The square root of 4 is 2, so we return 2. **Example 2: Input:** x = 8, **Output:** 2 **Explanation:** The square root of 8 is 2.82842..., and since we round it down to the nearest integer, 2 is returned.	
	13	* **Right Triangle Star Pattern**	**CO3**
	14	1 0 1 1 0 1 0 1 0 1 1 0 1 0 1	**CO3**
	14	* * *	**CO3**

.....
.....

		* * * * * * *	
	15	 Hollow Mirrored Right Triangle Star Pattern	co3
	16	 Hollow Diamond Star Pattern	co3

.....
.....

	17	 Right Arrow Star Pattern	co3
	18	 Left Arrow Star Pattern	co3
	19	WAP to print the pattern.... A B C D E F G G F E D C B A A B C D E F F E D C B A A B C D E E D C B A A B C D D C B A A B C C B A A B B A A A A	co3
	20	1 2 3 2 3 4 5 4 3	co3

.....
.....

		4 5 6 7 6 5 4 5 6 7 8 9 8 7 6 5 4 5 6 7 6 5 4 3 4 5 4 3 2 3 2 1	
	21	Given **N**, print the pattern as given in the example. **Input Format:** The first line of the input contains a single integer T denoting the number of test cases. First line of each test case contains one integer N. **Output Format:** For each test case, in a new line, print the given pattern as shown in the example. **Example 1:** Input: 1 4 Output: 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 2 3 3 3 2 1 1 2 3 4 3 2 1 1 2 3 3 3 2 1 1 2 2 2 2 2 1 1 1 1 1 1 1 1	**co3**
BIT MANIPULATION	1	Given a positive integer n, write a function to find if it is a power of 2 or not Examples: Input : n = 4, Output : Yes, Explanation: $2^2 = 4$	**co3**

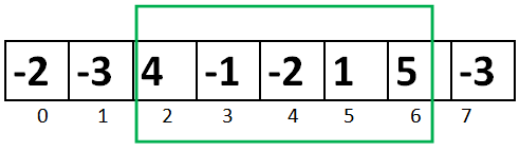
		Input : n = 42,Output : No, Explanation: 42 is not a power of 2 Input : n = 1,Output : Yes. Explanation: $2^0 = 1$	
	2	Given an integer n, calculate the square of a number without using *, / and pow(). Input: n = 5,Output: 25 Input: 7,Output: 49	**CO3**
	3	Given two integers say a and b. Find the quotient after dividing a by b without using multiplication, division, and mod operator. Example: Input: a = 10, b = 3, Output: 3 Input : a = 43, b = -8, Output : -5	**CO3**
	4	Given two numbers A and B. Write a program to count the number of bits needed to be flipped to convert A to B. Examples: Input: A = 10, B = 20, Output: 4 Explanation: Binary representation of A is 000**0101**0 Binary representation of B is 00010100 We need to flip highlighted four bits in A to make it B. Input: A = 7, B = 10, Output: 3 Explanation: Binary representation of A is 00000**111** Binary representation of B is 00001010 We need to flip highlighted three bits in A to make it B.	**CO3**
USER DEFINE FUNCTION	1	Write a program to print out all the Armstrong number between 100 and 500 using user define function.	**CO4**
	2	Write a program to print the entire prime no between 1 and 300 using user define function.	**CO4**
	3	Write a program to find sum of Fibonacci series up to N (where N is entered through keyboard) using function	**CO4**
	4	Write a program to find the value of y for a particular value of n. The a, x,b, n is input by user if n=1 $y=a^x$ % b	**CO4**

		if n=2 $y=ax^2+b^2$ if n=3 $y=a-b^x$ if n=4 $y=a+x^b$	
	5	Write a C function to test whether a given pair of numbers are amicable numbers. (Amicable number are pairs of numbers each of whose divisors add to the other)	CO4
RECURSION	1	Write a program to calculate the GCD of given numbers using recursion	CO4
	2	Write a program to find the sum of digits of a 5 digit number using recursion.	CO4
	3	Given a positive integer N, the task is to find the sum of the series 1 – (1/2) + (1/3) – (1/4) +…. (1/N) using recursion. Examples: Input: N = 3 Output: 0.8333333333333333 Explanation: 1 – (1/2) + (1/3) = 0.8333333333333333 Input: N = 4 Output: 0.5833333333333333 Explanation: 1- (1/2) + (1/3) – (1/4) = 0.5833333333333333	CO4
	4	**Modular exponentiation (Recursive)** Given three numbers a, b and c, we need to find (a^b) % c Now why do "% c" after exponentiation, because a^b will be really large even for relatively small values of a, b and that is a problem because the data type of the language that we try to code the problem, will most probably not let us store such a large number. Examples: Input : a = 2312 b = 3434 c = 6789 Output : 6343 Input : a = -3 b = 5 c = 89 Output : 24	CO4
ONE DIMENSION INTEGER ARRAY	1	Write a program, enter n elements into an array, perform linear search	CO4
	2	Write a program, enter n elements into an array, perform binary search.	CO4

.....
.....

	3	Write a program, enter n elements into an array, perform reverse operations	CO4
	4	Write a program to delete duplicate element in a list of 10 elements &display it on screen.	CO4
	5	Write a program to merge two sorted array & no element is repeated during merging.	CO4
	6	**Segregate 0s and 1s in an array** You are given an array of 0s and 1s in random order. Segregate 0s on left side and 1s on right side of the array [Basically you have to sort the array]. Traverse array only once. Input array = [0, 1, 0, 1, 0, 0, 1, 1, 1, 0] Output array = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]	CO4
	7	**Find Second largest element in an array** Given an array of integers, our task is to write a program that efficiently finds the second-largest element present in the array. **Examples:** ***Input:*** *arr[] = {12, 35, 1, 10, 34, 1}* ***Output:*** *The second largest element is 34.* ***Explanation:*** *The largest element of the array is 35 and the second largest element is 34* ***Input:*** *arr[] = {10, 5, 10}* ***Output:*** *The second largest element is 5.* ***Explanation:*** *The largest element of the array is 10 and the second largest element is 5*	CO4
	8	**Sort an array in wave form**	CO4

.....
.....

		Given an unsorted array of integers, sort the array into a wave array. An array **arr[0..n-1]** is sorted in wave form if: **arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= .....** **Examples:** ***Input:*** *arr[] = {10, 5, 6, 3, 2, 20, 100, 80}* ***Output:*** *arr[] = {10, 5, 6, 2, 20, 3, 100, 80}* ***Explanation:*** *here you can see {10, 5, 6, 2, 20, 3, 100, 80} first element is larger than the second and the same thing is repeated again and again. large element – small element-large element -small element and so on .it can be small element-larger element – small element-large element -small element too. all you need to maintain is the up-down fashion which represents a wave. there can be multiple answers.* ***Input:*** *arr[] = {20, 10, 8, 6, 4, 2}* ***Output:*** *arr[] = {20, 8, 10, 4, 6, 2}*	
	9	**Largest Sum Contiguous Subarray** Given an array **arr[]** of size **N**. The task is to find the sum of the contiguous subarray within a **arr[]** with the largest sum. **Largest Subarray Sum Problem**  **4 + (-1) + (-2) + 1 + 5 = 7** **Maximum Contiguous Array Sum is 7**	CO4
	10	**Sliding Window Max** You are given an array of integers nums, there is a sliding window of size k which is moving from the very left of the array to the very right. You can only see the k numbers in the	CO4

.....
.....

		window. Each time the sliding window moves right by one position. Return *the max sliding window*. Example 1 ***Input:*** *arr[] = {1, 2, 3, 1, 4, 5, 2, 3, 6}, K = 3* ***Output:*** *3 3 4 5 5 5 6* ***Explanation:*** *Maximum of 1, 2, 3 is 3* *Maximum of 2, 3, 1 is 3* *Maximum of 3, 1, 4 is 4* *Maximum of 1, 4, 5 is 5* *Maximum of 4, 5, 2 is 5* *Maximum of 5, 2, 3 is 5* *Maximum of 2, 3, 6 is 6* ***Example 2:*** ***Input****: arr[] = {8, 5, 10, 7, 9, 4, 15, 12, 90, 13}, K = 4* ***Output****: 10 10 10 15 15 90 90*	
	11	**Stock Buy Sell to Maximize Profit** The cost of a stock on each day is given in an array. Find the maximum profit that you can make by buying and selling on those days. If the given array of prices is sorted in decreasing order, then profit cannot be earned at all. **Examples:** ***Input:*** *arr[] = {100, 180, 260, 310, 40, 535, 695}* ***Output:*** *865* ***Explanation:*** *Buy the stock on day 0 and sell it on day 3 => 310 – 100 = 210* *Buy the stock on day 4 and sell it on day 6 => 695 – 40 = 655* *Maximum Profit = 210 + 655 = 865*	CO4

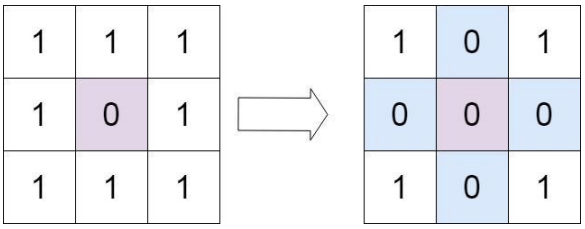
.....
.....

		Input: *arr[] = {4, 2, 2, 2, 4}* ***Output:*** *2* ***Explanation:*** *Buy the stock on day 1 and sell it on day 4 => 4 – 2 = 2* *Maximum Profit = 2* **Input:** prices = [7,1,5,3,6,4] **Output:** 5 **Explanation:** Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5. Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.	
	12	**TWO SUM** : Given an array of integers nums and an integer target, return *indices of the two numbers such that they add up to target*. You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice. You can return the answer in any order. **Example 1:** **Input:** nums = [2,7,11,15], target = 9,**Output:** [0,1] **Explanation:** Because nums[0] + nums[1] == 9, we return [0, 1]. **Example 2:** **Input:** nums = [3,2,4], target = 6,**Output:** [1,2] **Example 3:** **Input:** nums = [3,3], target = 6,**Output:** [0,1]	**CO4**
	13	**Triplet Sum** : Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.	**CO4**

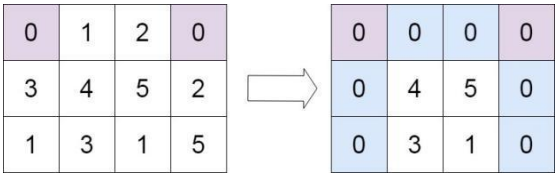
.....
.....

		Notice that the solution set must not contain duplicate triplets. **Example 1:** **Input:** nums = [-1,0,1,2,-1,-4] **Output:** [[-1,-1,2],[-1,0,1]] **Explanation:** nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0. nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0. nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0. The distinct triplets are [-1,0,1] and [-1,-1,2]. Notice that the order of the output and the order of the triplets does not matter. **Example 2:** **Input:** nums = [0,1,1] **Output:** [] **Explanation:** The only possible triplet does not sum up to 0. **Example 3:** **Input:** nums = [0,0,0] **Output:** [[0,0,0]] **Explanation:** The only possible triplet sums up to 0.	
	14	**Chocolate Distribution Problem** Given an array A[ ] of positive integers of size N, where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are	CO4

……
……

		M students, the task is to distribute chocolate packets among M students such that : 1. Each student gets exactly one packet. 2. The difference between maximum number of chocolates given to a student and minimum number of chocolates given to a student is minimum.	
TWO DIMENSION INTEGER ARRAY	1	Write a program to evaluate the addition of diagonal elements of two Square matrices.	**CO4**
	2	Write a program to find the transpose of a given matrix & check whether it is symmetric or not.	**CO4**
	3	Write a program to print the multiplication of two N*N (Square) matrix.	**CO4**
	4	Print all elements in sorted order from row and column wise sorted matrix Given an n x n matrix, where every row and column is sorted in non-decreasing order. Print all elements of the matrix in sorted order. Example: Input: mat[][] = { {10, 20, 30, 40}, {15, 25, 35, 45}, {27, 29, 37, 48}, {32, 33, 39, 50}, }; Output: 10 15 20 25 27 29 30 32 33 35 37 39 40 45 48 50	**CO4**
	5	**Set Matrix Zeros** Given an m x n integer matrix matrix, if an element is 0, set its entire row and column to 0's. **Example 1:**  Input: matrix = [[1,1,1],[1,0,1],[1,1,1]] Output: [[1,0,1],[0,0,0],[1,0,1]]	**CO4**

……
……

		Example 2:  Input: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]] Output: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]	
STRING	1	Write a program in C to check whether the given string is a palindrome or not.	**CO4**
	2	Create a user define function replaceCharacter() that replace character from the string. String : This book is very easy Character : s replace character : p Output : Thip book ip very eapy	**CO4**
	3	Write a function reverseStr(char str[]) that takes as argument and reverse it. String :- amit kumar Reverse String :- ramuk tima	**CO4**
	4	Write a function reverseWordLetter(char str[]) that takes as an argument and reverse it. String : I love my India Reverse String : I evol ym aidnI	**CO4**
	5	Write a function reverseWord(char str[]) that takes as an argument and reverse it.	**CO4**

.....
.....

		String : I love my India Reverse String: India my love I	
	6	Write a function which delete all the repeated occurrences of a character from a string. String : This book is very easy Output : This bok very a	co4
	7	Write program to sort the array of character (String) in alphabetical order like STRING in GINRST.	co4
	8	Write a program to remove all the blank space from the string & print it, also count the no of characters.	Co4
	9	Print all the duplicate characters in a string Given a string S, the task is to print all the duplicate characters with their occurrences in the given string. Example: Input: S = “goodorgood” Output: d, count = 2 g, count = 2 o, count = 5 r, count = 1	Co4
	10	**LONGEST SUBSTRING WITHOUT REPEATING CHARACTERS** Given a string s, find the length of the **longest substring** without repeating characters. **Example 1:** **Input:** s = "abcabcbb",**Output:** 3 **Explanation:** The answer is "abc", with the length of 3. **Example 2:** **Input:** s = "bbbbb",**Output:** 1	Co4

.....
.....

		Explanation: The answer is "b", with the length of 1. **Example 3:** **Input:** s = "pwwkew",**Output:** 3 **Explanation:** The answer is "wke", with the length of 3. Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.	
	11	**Permutations of given String** Given a string S, the task is to write a program to print all permutations of a given string. A permutation also called an “arrangement number” or “order,” is a rearrangement of the elements of an ordered list S into a one-to-one correspondence with S itself. A string of length N has N! permutations. Examples: Input: S = “ABC” Output: “ABC”, “ACB”, “BAC”, “BCA”, “CBA”, “CAB” Input: S = “XY” Output: “XY”, “YX”	Co4
	12	Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in **any order**. A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.	Co4

.....
.....

		  Example 1: **Input:** digits = "23" **Output:** ["ad","ae","af","bd","be","bf","cd","ce","cf"] **Example 2:** **Input:** digits = "" **Output:** [] **Example 3:** **Input:** digits = "2" **Output:** ["a","b","c"]	
	13	**Wildcard Pattern Matching** Given a text and a wildcard pattern, implement wildcard pattern matching algorithm that finds if wildcard pattern is matched with text. The matching should cover the entire text (not partial text). The wildcard pattern can include the characters '?' and '*' '?' – matches any single character '*' – Matches any sequence of characters (including the empty sequence) For example: Text = "baaabab", Pattern = "*****ba*****ab", output : true Pattern = "baaa?ab", output : true Pattern = "ba*a?", output : true Pattern = "a*ab", output : false	**Co4**
	14	**Naive algorithm for Pattern Searching**	**Co4**

……
……

		Given text string with length n and a pattern with length m, the task is to prints all occurrences of pattern in text. Note: You may assume that n > m. Examples: Input: text = "THIS IS A TEST TEXT", pattern = "TEST" Output: Pattern found at index 10 Input: text = "AABAACAADAABAABA", pattern = "AABA" Output: Pattern found at index 0, Pattern found at index 9, Pattern found at index 12	
STRUCTURE & UNION	1	Write a union program to extract individual bytes from an unsigned int. (for hexadecimal value)	**Co4**
	2	Define a structure that can describe a hotel. It should have the member that includes the name, address, grade, room charge and number of rooms. Write a function to print out hotel of given grade in order of room charges.	**Co4**
	3	Define a structure called cricket with player name, team name, batting average, for 50 players & 5 teams. Print team wise list contains names of player with their batting average.	**Co4**
FILE HANDLING	1	Write a c program to copy & count the character content of one file says a.txt to another file b.txt.	**Co5**
	2	Write a program to take 10 integers from file and write square of these integer in other file.	**Co5**
	3	Write a program to read number from file and then write all 'odd' number to file ODD.txt & all even to file EVEN.txt.	**Co5**
	4	Write a program to print all the prime number, between 1 to 100 in file Prime.txt.	**Co5**
GRAPHICAL OPERATIONS	1.	Write a program to draw circle, line, rectangle fill with red color.	**Co5**

……
……

	2.	Draw a moving cycle using computer graphics programming in C.	**Co5**

Beyond the Syllabus

Category	Program No.	Title
Command Line Argument	1	Write a program to find the factorial of given number using command line argument.
	2	Write a program to find the sum of digits of a 5 digit number using command line argument.
Dynamic Programming	1	Write a C program to print the n-th Fibonacci Number
	2	**Longest Common Subsequence (LCS)** Given two strings, S1 and S2, the task is to find the length of the Longest Common Subsequence. If there is no common subsequence, return 0. A subsequence is a string generated from the original string by deleting 0 or more characters and without changing the relative order of the remaining characters. For example , subsequences of "ABC" are "", "A", "B", "C", "AB", "AC", "BC" and "ABC". In general a string of length n has 2n subsequences. LCS problem has great applications like diff utility (find the difference between two files) that we use in our day to day software development. Examples: Input: S1 = "ABC", S2 = "ACD" Output: 2 Explanation: The longest subsequence which is present in both strings is "AC". Input: S1 = "AGGTAB", S2 = "GXTXAYB"

……
……

		Output: 4 Explanation: The longest common subsequence is “GTAB”. Input: S1 = “ABC”, S2 = “CBA” Output: 1 Explanation: There are three common subsequences of length 1, “A”, “B” and “C” and no common subsequence of length more than 1.