CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

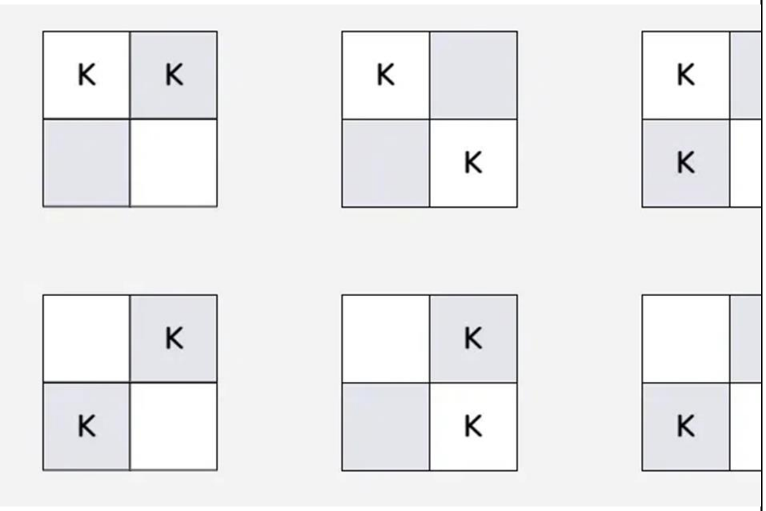
Faculty of Technology and Engineering

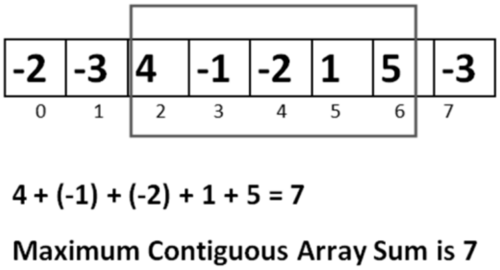
CE / IT/ CSE/ AI-ML

Practical List

Academic Year	:	2025-26	Semester	:	5th
Course code	:	CSE311	Course name	:	Competitive programming essentials

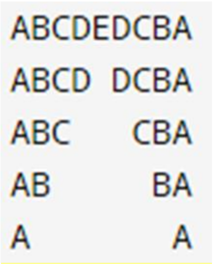
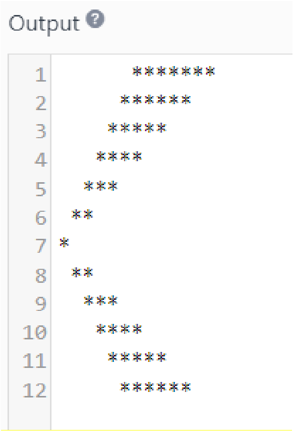
Sr. No.	Aim		Hours	CO
1.	Programming techniques		02	1
	1.1	Calculate the number of trailing zeros in the factorial n! For example, 20! =2432902008176640000 and it has 4 trailing zeros. **Input** The only input line has an integer n. **Output** Print the number of trailing zeros in n! **Constraints** 1 <= n <= 10^9 **Example** Input:20 Output:4		

	1.2	Your task is to count for k=1, 2…, n the number of ways two knights can be placed on a k×k chessboard so that they do not attack each other. For Example: Input: k = 2 Output: (for 1×1 and 2×2) 0 6  **Input** The only input line contains an integer n. **Output** Print n integers: the results. **Constraints** 1 <= q <= 10000 Example Input: 8 **Output** (from 1×1 to 8×8): 0 6 28 96 252 550 1056 1848		
2.	**Time efficiency and situational awareness**			

	2.1	**Maximum subarray sum problem** https://leetcode.com/problems/maximum-subarray/description/ Given an integer array nums, find the subarray with the largest sum, and return its sum.  -2 -3 4 -1 -2 1 5 -3 0 1 2 3 4 5 6 7 4 + (-1) + (-2) + 1 + 5 = 7 Maximum Contiguous Array Sum is 7 Example 1: Input: nums = [-2,1,-3,4,-1,2,1,-5,4] Output: 6 Explanation: The subarray [4,-1,2,1] has the largest sum 6. Example 2: Input: nums = [1] Output: 1 Explanation: The subarray [1] has the largest sum 1. Example 3: Input: nums = [5,4,-1,7,8] Output: 23 Explanation: The subarray [5,4,-1,7,8] has the largest sum 23. **Followup**: **Maximum circular subarray sum** https://leetcode.com/problems/maximum-sum-circular-subarray/description/ Given a circular integer array nums of length n, return the maximum possible sum of a non-empty subarray of nums. A circular array means the end of the array connects to the beginning of the array. Formally, the next element of nums[i] is nums[(i + 1) % n] and the previous element of nums[i] is nums[(i - 1 + n) % n]. A subarray may only include each element of the fixed buffer nums at most once. Formally, for a subarray nums[i], nums[i + 1], ..., nums[j], there does not exist i <= k1, k2 <= j with k1 % n == k2 % n. Example:	02

		Input: nums = [5,-3,5] Output: 10 Explanation: Subarray [5,5] has maximum sum 5 + 5 = 10.	

	2.2	**SORT COLORS** https://leetcode.com/problems/sort-colors/description/ Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively. You must solve this problem without using the library's sort function. Example: Input: nums = [2,0,2,1,1,0] Output: [0,0,1,1,2,2]	

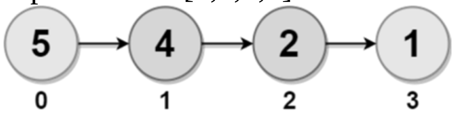
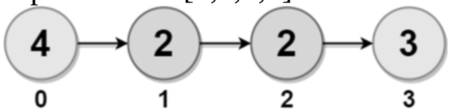
	1.3	Pattern problems + bit manipulation (a) Alphabet archN = 5  ABCDEDCBA ABCD DCBA ABC CBA AB BA A A (b) left arrow N = 7  Output ******* ****** ***** **** *** ** * ** *** **** ***** ******	02	
3.	Bit manipulation			4
	3.1	Given a limited range array of size n and containing elements between 1 and n+1 with one element missing, find the missing number without using any extra space. Example: Input: {3, 2, 4, 6, 1} Output: The missing element is 5 Input: {3, 2, 4, 5, 6} Output: The missing element is 1 Input: {3, 2, 4, 5, 1} Output: The missing element is 6		
	3.2	Given an unsorted array where every number appears an even number of times except for two numbers that appear an odd number of times, identify the two numbers with		

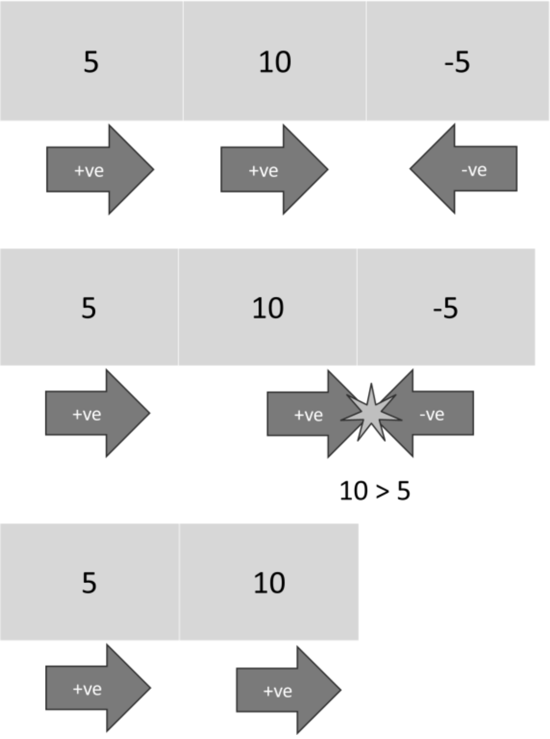
		odd occurrences. This must be done with a time complexity of O(n) and using only O(1) extra space. Examples: Input: {3, 7, 9, 3, 3, 7, 3, 5} Output: 9 and 5 Input: {8, 8, 15, 1024, 8, 8, 8, 8, 15, 15} Output: 15 and 1024 Input: {25, 35} Output: 25 and 35		
4.	Recursion		02	1

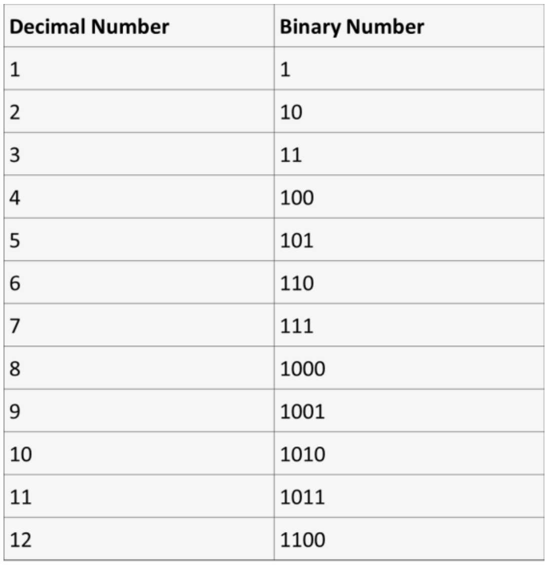
	4.1	Generating Subsets:
		Given an array of size N, print all the subsets of the array.
		Example:
		Input: N = 3, Array = [1, 2, 3]
		Output: {}
		{1}
		{1, 2}
		{1, 2, 3}
		{1, 3}
		{2}
		{2, 3}
		{3}		

	4.2	Generating permutations using backtracking: You are given with the string as an input and need to print all the permutation of the given input string. **Example**: Input: ABC Output: ABC ACB BCA BAC CAB CBA		
5.	Linear data structures (Array, linkedlist)			
	5.1	Determine if a 9 x 9 Sudoku board is valid. Only the filled cells need to be validated according to the following rules:	02	

		Each row must contain the digits 1-9 without repetition. Each column must contain the digits 1-9 without repetition. Each of the nine 3 x 3 sub-boxes of the grid must contain the digits 1-9 without repetition. Note: A Sudoku board (partially filled) could be valid but is not necessarily solvable. Only the filled cells need to be validated according to the mentioned rules. Please note that a “0” denotes an empty cell Examples: Input: 1 4 7 0 0 0 0 0 3 2 5 0 0 0 1 0 0 0 3 0 9 0 0 0 0 0 0 0 8 0 0 2 0 0 0 4 0 0 0 4 1 0 0 2 0 9 0 0 0 0 0 6 0 0 0 0 3 0 0 0 0 0 9 4 0 0 0 0 2 0 0 0 0 0 1 0 0 8 0 0 7 Output: True Input: 1 4 4 0 0 0 0 0 3 2 5 0 0 0 1 0 0 0 3 0 9 0 0 0 0 0 0 0 8 0 0 2 0 0 0 4 0 0 0 4 1 0 0 2 0 9 0 0 0 0 0 6 0 0 0 0 3 0 0 0 0 0 9 4 0 0 0 0 2 0 0 0 0 0 1 0 0 8 0 0 7 Output: False		
	5.2	**Maximum Twin Sum of a Linked List** In a linked list of size n, where n is even, the ith node (0-indexed) of the linked list is known as the twin of the (n-1-i)th node, if 0 <= i <= (n / 2) - 1.	02	

		For example, if n = 4, then node 0 is the twin of node 3, and node 1 is the twin of node 2. These are the only nodes with twins for n = 4. The twin sum is defined as the sum of a node and its twin. Given the head of a linked list with even length, return the maximum twin sum of the linked list. **Example 1:** Input: head = [5,4,2,1]  Output: 6 Explanation: Nodes 0 and 1 are the twins of nodes 3 and 2, respectively. All have twin sum = 6. There are no other nodes with twins in the linked list. Thus, the maximum twin sum of the linked list is 6. **Example 2:** Input: head = [4,2,2,3]  Output: 7 Explanation: The nodes with twins present in this linked list are: - Node 0 is the twin of node 3 having a twin sum of 4 + 3 = 7. - Node 1 is the twin of node 2 having a twin sum of 2 + 2 = 4. Thus, the maximum twin sum of the linked list is max(7, 4) = 7.		
6.	Linear data structures (stack, queue)		02	
	6.1	We are given an array: asteroids of integers representing asteroids in a row. For each asteroid, the absolute value represents its size, and the sign represents its direction (positive meaning right, negative meaning left). Each asteroid moves at the same speed. Find out the state of the asteroids after all collisions. If two asteroids meet, the smaller one will explode. If both		

		are the same size, both will explode. Two asteroids moving in the same direction will never meet. Example 1: Input: asteroids = [8,-8] Output: [] Explanation: The 8 and -8 collide exploding each other. Example 2: Input: asteroids = [5,10,-5] Output: [5,10] Explanation: The 10 and -5 collide resulting in 10. The 5 and 10 never collide.  5 10 -5 +ve +ve -ve 5 10 -5 +ve +ve -ve 10 > 5 5 10 +ve +ve		
	6.2	**Generate Binary Numbers from 1 to n using queue and its operations.** Given a number N, write a function that generates and prints all binary numbers with decimal values from 1 to N. **Example 1:** Input: n = 2 Output: 1, 10 **Example 2:** Input: n = 5 Output: 1, 10, 11, 100, 101		



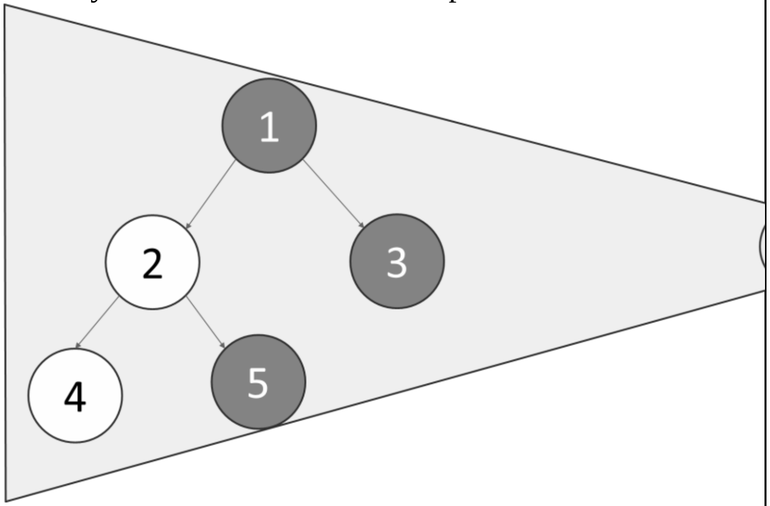
Decimal Number	Binary Number
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100

7. Non-Linear data structures (BT, Map, Graph(basic))

7.1

Binary Tree Right View

Given the root of a binary tree, imagine yourself standing on the right side of it, returning the values of the nodes which you can see ordered from top to bottom.



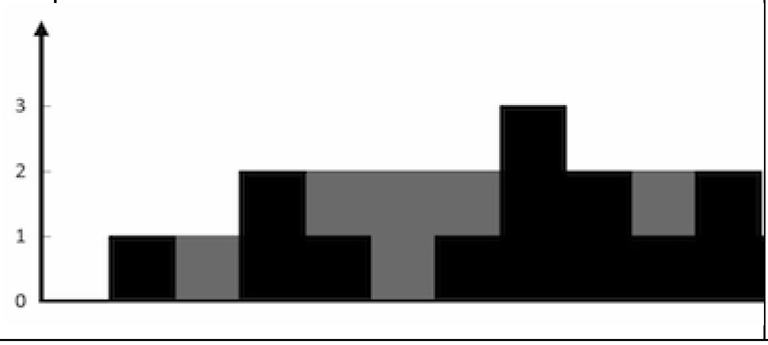
Example 1:
Input: root = [1,2,3,4,5,null,null]
Output: [1,3,5]

Example 2:
Input: root = [1,null,3]
Output: [1,3]

02

		Example 3: Input: root = [] Output: []		
	7.2	Given an array of integers arr, return true if the number of occurrences of each value in the array is unique or false otherwise. Example 1: Input: arr = [1,2,2,1,1,3] Output: true Explanation: The value 1 has 3 occurrences, 2 has 2 and 3 has 1. No two values have the same number of occurrences. Example 2: Input: arr = [1,2] Output: false Example 3: Input: arr = [-3,0,1,-3,1,1,1,-3,10,0] Output: true Constraints: 1 <= arr.length <= 1000 -1000 <= arr[i] <= 1000	02	
	7.3	In a town, there are n people labeled from 1 to n. There is a rumor that one of these people is secretly the town judge. If the town judge exists, then: ● The town judge trusts nobody. ● Everybody (except for the town judge) trusts the town judge. ● There is exactly one person that satisfies properties 1 and 2. You are given an array trust where trust[i] = [ai, bi] representing that the person labeled ai trusts the person labeled bi. If a trust relationship does not exist in the trust array, then such a trust relationship does not exist. Return the label of the town judge if the town judge exists and can be identified, or return -1 otherwise.	02	

		Example 1: Input: n = 2, trust = [[1,2]] Output: 2 Example 2: Input: n = 3, trust = [[1,3],[2,3]] Output: 3 Example 3: Input: n = 3, trust = [[1,3],[2,3],[3,1]] Output: -1 Constraints: 1 <= n <= 1000 0 <= trust.length <= 104 trust[i].length == 2 All the pairs of trust are unique. ai != bi 1 <= ai, bi <= n		
8.	Searching and sorting			
	8.1	Ferris wheel: There are N children who want to go to a Ferris wheel in the form of an array arr[], and your task is to find a gondola for each child. Each gondola may have one or two children in it, and in addition, the total weight in a gondola may not exceed X. You know the weight of every child. What is the minimum number of gondolas needed for the children? Input: N = 4, X = 10, arr[] = {7, 2, 3, 9} Output: 3 Explanation: We need only 3 gondolas: {2, 3}, {7} and {9}. Input: N = 4, X = 6, arr[] = {2, 3, 3, 4} Output: 2 Explanation: We need only 2 gondolas: {2, 4} and {3, 3}	02	
	8.2	**Sort Vowels in a String** Given a 0-indexed string s, permute s to get a new string t such that: All consonants remain in their original places. More formally, if there is an index i with 0 <= i < s.length such that s[i] is a consonant, then t[i] = s[i].	02	

		The vowels must be sorted in the nondecreasing order of their ASCII values. More formally, for pairs of indices i, j with 0 <= i < j < s.length such that s[i] and s[j] are vowels, then t[i] must not have a higher ASCII value than t[j]. Return the resulting string. The vowels are 'a', 'e', 'i', 'o', and 'u', and they can appear in lowercase or uppercase. Consonants comprise all letters that are not vowels. **Example 1**: Input: s = "lEetcOde" Output: "lEOtcede" Explanation: 'E', 'O', and 'e' are the vowels in s; 'l', 't', 'c', and 'd' are all consonants. The vowels are sorted according to their ASCII values, and the consonants remain in the same places. **Example 2**: Input: s = "lYmpH" Output: "lYmpH" Explanation: There are no vowels in s (all characters in s are consonants), so we return "lYmpH". **Constraints**: 1 <= s.length <= 10^5 s consists only of letters of the English alphabet in uppercase and lowercase.		
9	**Algorithm design**			
	9.1	**Two pointers**: Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining. Example 1: Input: height = [0,1,0,2,1,0,1,3,2,1,2,1] Output: 6 	02	

		Explanation: The above elevation map (black section) is represented by an array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped. Example 2: Input: height = [4,2,0,3,2,5] Output: 9 Constraints: n == height.length 1 <= n <= 2 * 10^4 0 <= height[i] <= 10^5		
	9.2	**Next greater/smaller element:** Given an array of integers temperatures represents the daily temperatures, return an array answer such that answer[i] is the number of days you have to wait after the ith day to get a warmer temperature. If there is no future day for which this is possible, keep answer[i] == 0 instead. Example 1: Input: temperatures = [73,74,75,71,69,72,76,73] Output: [1,1,4,2,1,1,0,0] Example 2: Input: temperatures = [30,40,50,60] Output: [1,1,1,0] Example 3: Input: temperatures = [30,60,90] Output: [1,1,0]	02	
	9.3	**Sliding window:** You have a bomb to defuse, and your time is running out! Your informer will provide you with a circular array code of length of n and a key k. To decrypt the code, you must replace every number. All the numbers are replaced simultaneously. If k > 0, replace the ith number with the sum of the next k numbers. If k < 0, replace the ith number with the sum of the previous k numbers. If k == 0, replace the ith number with 0.	02	

		As code is circular, the next element of code[n-1] is code[0], and the previous element of code[0] is code[n-1]. Given the circular array code and an integer key k, return the decrypted code to defuse the bomb! **Example 1:** Input: code = [5,7,1,4], k = 3 Output: [12,10,16,13] Explanation: Each number is replaced by the sum of the next 3 numbers. The decrypted code is [7+1+4, 1+4+5, 4+5+7, 5+7+1]. Notice that the numbers wrap around. **Example 2:** Input: code = [1,2,3,4], k = 0 Output: [0,0,0,0] Explanation: When k is zero, the numbers are replaced by 0. **Example 3:** Input: code = [2,4,9,3], k = -2 Output: [12,5,6,13] Explanation: The decrypted code is [3+9, 2+3, 4+2, 9+4]. Notice that the numbers wrap around again. If k is negative, the sum is of the previous numbers. **Constraints**: n == code.length 1 <= n <= 100 1 <= code[i] <= 100 -(n - 1) <= k <= n - 1		
10.	String processing			
	10.1	**Wildcard matching:** Given an input string (s) and a pattern (p), implement wildcard pattern matching with support for '?' and '*' where: '?' Matches any sing8le character. '*' Matches any sequence of characters (including the empty sequence). The matching should cover the entire input string (not partial).	02	

		Example 1: Input: s = "aa", p = "a" Output: false Explanation: "a" does not match the entire string "aa". **Example 2**: Input: s = "aa", p = "*" Output: true Explanation: '*' matches any sequence. **Example 3**: Input: s = "cb", p = "?a" Output: false Explanation: '?' matches 'c', but the second letter is 'a', which does not match 'b'. **Constraints**: 0 <= s.length, p.length <= 2000 s contains only lowercase English letters. p contains only lowercase English letters, '?' or '*'.		
	10.2	**Find Longest Awesome Substring** You are given a string s. An awesome substring is a non-empty substring of s such that we can make any number of swaps in order to make it a palindrome. Return the length of the maximum length awesome substring of s. **Example 1:** Input: s = "3242415" Output: 5 Explanation: "24241" is the longest awesome substring, we can form the palindrome "24142" with some swaps. **Example 2:** Input: s = "12345678" Output: 1 **Example 3:** Input: s = "213123" Output: 6 Explanation: "213123" is the longest awesome substring, we can form the palindrome "231132" with some swaps. **Constraints**: 1 <= s.length <= 105 s consists only of digits.		