Faculty of Technology and Engineering

U & P U. Patel Department of Computer Engineering

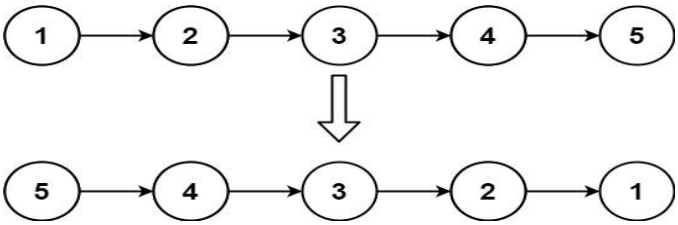
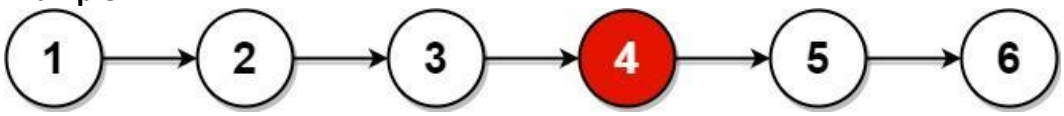
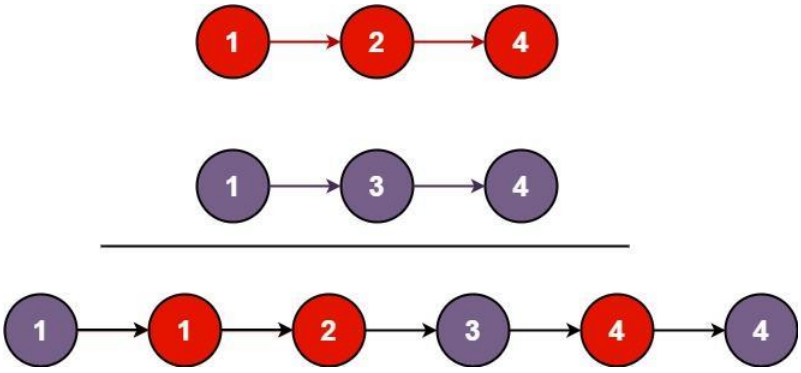
Date: 20/06/2023

Practical List

Academic Year	:	2023-24	Semester	:	3rd
Course code	:	CE261	Course name	:	Data Structures & Algorithms

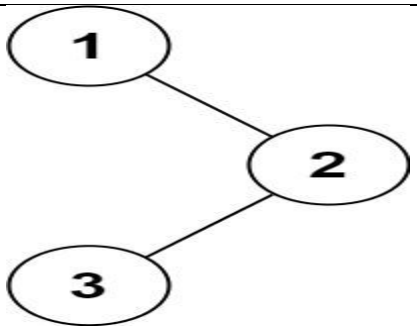
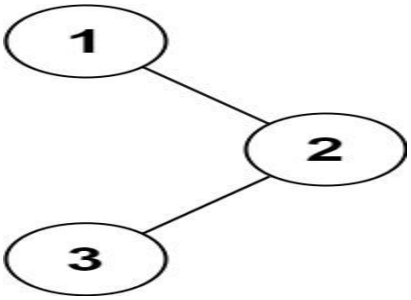
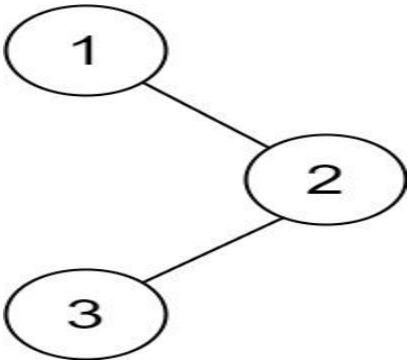
Sr. No.	Aim	Hours	CO
1	Installation of VS Code. Implement Linear Search and Binary Search using array data structure.	02	1, 2
2	In a far away Galaxy of Tilky Way, there was a planet Tarth where the sport of Competitive Coding was very popular. According to legends, there lived a setter known for loving knapsack type problems. Given N objects in a row, with weights W1,W2,…,WN, you need to find the maximum number of consecutive objects you can fill in a bag of maximum capacity C such that the total weight of objects taken is at least K. In other words, pick objects such that-The total weight of collected objects is at least K. The total weight does not exceed C. The objects picked must be consecutive (i.e. a subarray of the objects need to be picked) The number of objects is maximized. You need to print this maximum value. **Note: If no such object could be picked, then the answer is obviously 0.** Input: • The first line of input contains T, number of test cases in a file. • The next line contains three integers, N, C and K, as described in the problem statement. • The next line contains N space separated integers, denoting Wi, i.e. weight of the object. Output: For test case, maximum number of objects you can pick. Input 2 5 5 5 5 4 3 2 1 5 5 4 1 4 1 1 1 Output 2 2	02	1,5

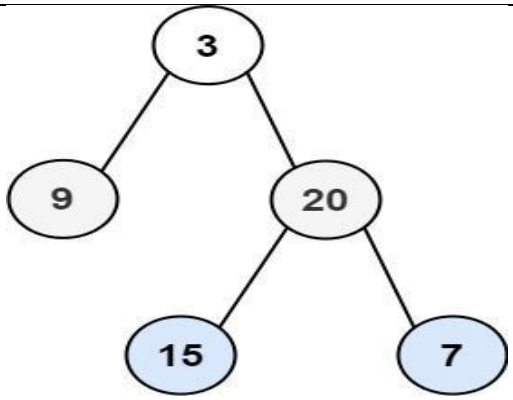
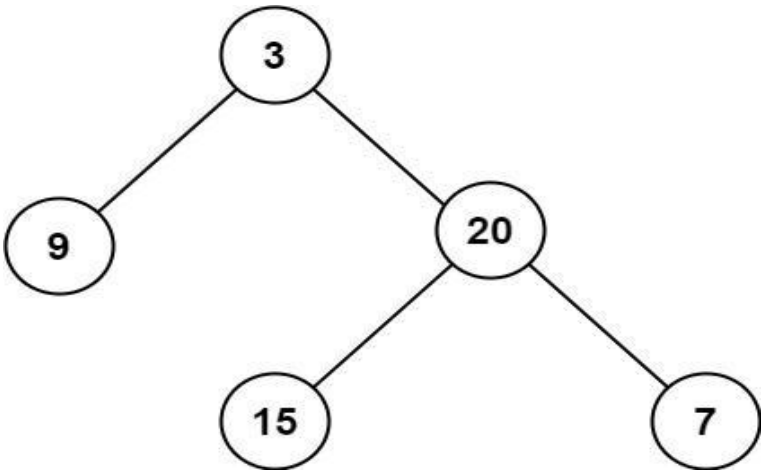
3	Implement Sorting Algorithm(s). (a) Bubble Sort (b) Selection Sort (c) Insertion Sort	04	1,2,6
4	Sort Colors: Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively. **Note: You must solve this problem without using the library's sort function.**	02	1,5,6
5	Chef and his little brother are playing with sticks. They have total N sticks. Length of i-th stick is Ai. Chef asks his brother to choose any four sticks and to make a rectangle with those sticks its sides. Chef warns his brother to not to break any of the sticks, he has to use sticks as a whole. Also, he wants that the rectangle formed should have the maximum possible area among all the rectangles that Chef's brother can make. Chef's little brother takes this challenge up and overcomes it. Can you also do so? That is, you have to tell whether it is even possible to create a rectangle? If yes, then you have to tell the maximum possible area of rectangle. Input • The first line contains a single integer T denoting the number of test-cases. T test cases follow. • The first line of each test case contains a single integer N denoting the number of sticks. • The second line of each test case contains N space-separated integers A1, A2, ..., AN denoting the lengths of sticks. Output • For each test case, output a single line containing an integer representing the maximum possible area for rectangle or -1 if it's impossible to form any rectangle using the available sticks. Input 2 5 1 2 3 1 2 4 1 2 2 3	02	1,5,6
6	Implement below operations of singly linked list. (a) Insert a node at front (b) Delete a node at last (c) Delete all nodes of linked list **Note: Display content of linked list after each operation.**	02	1,2
7	**Reverse Linked List** Given the head of a singly linked list, reverse the list, and return the reversed list. Input: head = [1,2,3,4,5] Output: [5,4,3,2,1]	02	1,2,5

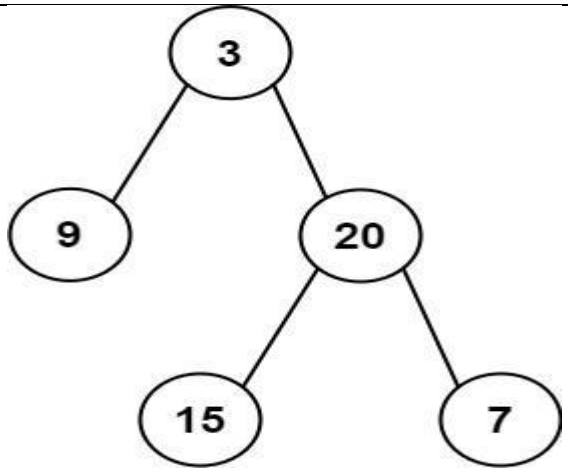
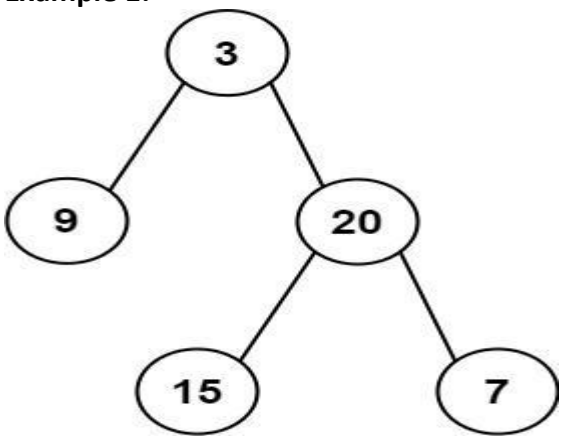
			
8	**Middle of the Linked List** Given the head of a singly linked list, return the middle node of the linked list. If there are two middle nodes, return the second middle node. **Example 1:**  Input: head = [1,2,3,4,5] Output: [3,4,5] Explanation: The middle node of the list is node 3. **Example 2:**  Input: head = [1,2,3,4,5,6] Output: [4,5,6] Explanation: Since the list has two middle nodes with values 3 and 4, we return the second one.	02	1,2
9	**Merge Two Sorted Lists** You are given the heads of two sorted linked lists list1 and list2. Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists. Return the head of the merged linked list. **Example 1:**  Input: list1 = [1,2,4], list2 = [1,3,4] Output: [1,1,2,3,4,4]	02	1,2,5
10	**Implement stack using array** Implement a program to implement a Stack using Array. Your task is to use the	02	1,2,3

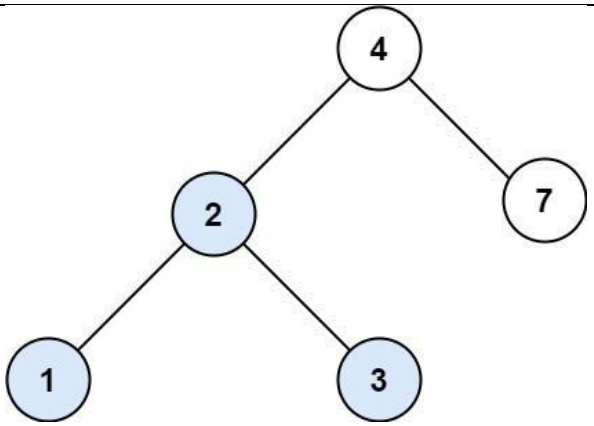
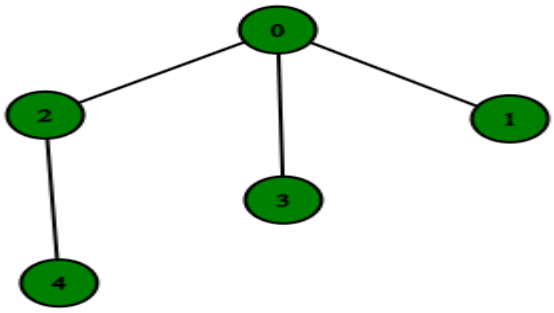
	class as shown in the comments in the code editor and complete the functions push () and pop () to implement a stack. **Example 1**: Input: push(2) push(3) pop() push(4) pop() Output: 3, 4		
11	**Implement Queue using array** Implement a Queue using an Array. Queries in the Queue are of the following type: (i) 1 x (a query of this type means pushing 'x' into the queue) (ii) 2 (a query of this type means to pop element from queue and print the poped element) **Example 1:** Input: Q = 5 Queries = 1 2 1 3 2 1 4 2 Output: 2 3	02	1,2,3
12	**Implement Stack using Linked List** You have a linked list and you have to implement the functionalities push and pop of stack using this given linked list. Your task is to use the class as shown in the comments in the code editor and complete the functions push () and pop () to implement a stack. **Example 1**: **Input:** push(2) push(3) pop() push(4) pop() **Output: 3 4**	02	1,2,3
13	**Implement Queue using Linked List** A Query Q is of 2 Types (i) 1 x (a query of this type means pushing 'x' into the queue) (ii) 2 (a query of this type means to pop an element from the queue and print the popped element) **Example 1:** Input: Q = 5 Queries = 1 2 1 3 2 1 4 2 Output: 2 3	02	1,2,3
14	**Valid Parentheses** Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.	02	1,2,5

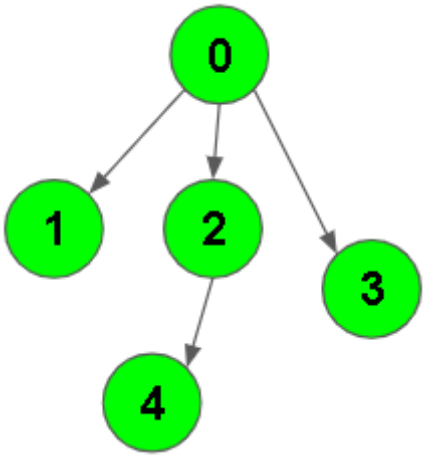
	An input string is valid if: Open brackets must be closed by the same type of brackets. Open brackets must be closed in the correct order. Every close bracket has a corresponding open bracket of the same type. **Example 1:** **Input: s = "()"** **Output: true**		
15	Chef has a sequence A1, A2, … … … AN and an integer K. Now there is a sliding window of size K which is moving from the very left of the array to the very right and at a particular time Chef has access to only those elements that are present in that window and Chef wants to find the number of the distinct elements of each window of size K. Help Chef to find the answer. Input The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows. The first line of each test case contains two integers N and K. The second line contains N space-separated integers A1,A2,A3,…..A_N Output For each test case, print a single line containing space-separated integers — the number of the distinct elements of each window of size from the very left of the array to the very right of the sequence. Example Input 2 6 3 5 8 5 4 4 1 4 2 4 5 2 2 Output: 2 3 2 2 2 2 1 Explanation: Example case 1: Number of the distinct elements of [5, 8, 5], [8, 5, 4] [5, 4, 4], [4, 4, 1] are respectively 2, 3, 2, 2.	02	1,2,5
16	**Binary Tree Inorder Traversal** Given the root of a binary tree, return the inorder traversal of its nodes' values. **Example 1:**	02	1,2

	 Input: root = [1, null,2,3] Output: [1,3,2]		
17	**Binary Tree Preorder Traversal** Given the root of a binary tree, return the preorder traversal of its nodes' values. **Example 1:**  Input: root = [1, null,2,3]	02	1,2
18	**Binary Tree Postorder Traversal** Given the root of a binary tree, return the postorder traversal of its nodes' values. **Example 1:**  Input: root = [1, null,2,3] Output: [3,2,1]	02	1,2
19	**Binary Tree Level Order Traversal** Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level). **Example 1:**	02	1,2,4

	 Input: root = [3,9,20, null, null,15,7] Output: [[3],[9,20],[15,7]]		
20	**Maximum Depth of Binary Tree** Given the root of a binary tree, return its maximum depth. A binary tree's maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node. **Example 1:**  Input: root = [3,9,20, null, null,15,7] Output: 3	02	1,2,4
21	**Construct Binary Tree from Preorder and Inorder Traversal** Given two integer arrays preorder and inorder where preorder is the preorder traversal of a binary tree and inorder is the inorder traversal of the same tree, construct and return the binary tree. **Example 1:**	02	1,2

	 Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7] Output: [3,9,20,null,null,15,7]		
22	**Construct Binary Tree from Inorder and Postorder Traversal** Given two integer arrays inorder and postorder where inorder is the inorder traversal of a binary tree and postorder is the postorder traversal of the same tree, construct and return the binary tree. **Example 1:**  **Input:** inorder = [9,3,15,20,7], postorder = [9,15,7,20,3] **Output:** [3,9,20,null,null,15,7]	02	1,2
23	**Search in a Binary Search Tree** You are given the root of a binary search tree (BST) and an integer val. Find the node in the BST that the node's value equals val and return the subtree rooted with that node. If such a node does not exist, return null. **Example 1:**	04	1,2,4

	 Input: root = [4,2,7,1,3], val = 2 Output: [2,1,3]		
24	**DFS of Graph** You are given a connected undirected graph. Perform a Depth First Traversal of the graph. **Note:** Use a recursive approach to find the DFS traversal of the graph starting from the 0th vertex from left to right according to the graph. **Example 1:** **Input: V = 5 , adj = [[2,3,1] , [0], [0,4], [0], [2]]**  Output: 0 2 4 3 1 Explanation: 0 is connected to 2, 3, 1. 1 is connected to 0. 2 is connected to 0 and 4. 3 is connected to 0. 4 is connected to 2. so starting from 0, it will go to 2 then 4, and then 3 and 1. Thus dfs will be 0 2 4 3 1.	04	1,2,4,5
25	**BFS of graph** Given a directed graph. The task is to do Breadth First Traversal of this graph starting from 0. **Note:** One can move from node u to node v only if there's an edge from u to v and	04	1,2,4,5

	find the BFS traversal of the graph starting from the 0th vertex, from left to right according to the graph. Also, you should only take nodes directly or indirectly connected from Node 0 in consideration. **Example 1:** **Input:**  Output: 0 1 2 3 4 Explanation: 0 is connected to 1 , 2 , 3. 2 is connected to 4. so starting from 0, it will go to 1 then 2 then 3.After this 2 to 4, thus bfs will be 0 1 2 3 4.		
26.	In an array of 20 elements, arrange 15 different values, which are generated randomly between 1,00,000 to 9,99,999. Use hash function to generate key using linear probing, quadratic probing and double hashing to avoid collision. H(k) = 2k + 3 and m = 20. Write a program to input and display the final values of array.	02	1,2,6