University of Mumbai

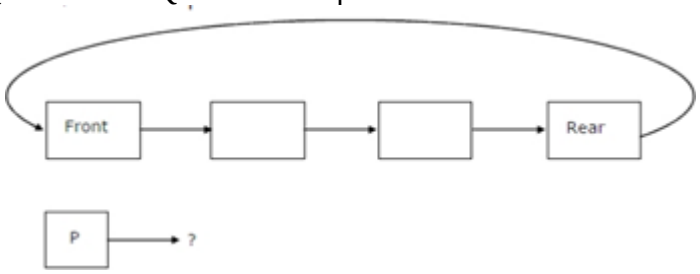
Program: **Computer Engineering**
Curriculum Scheme: Rev2019
Examination: Second Year Semester: III

Course Code: CSC303 Course Name: Data Structures

Time: 2 hour Max. Marks: 80

==========

Q1. All questions compulsory 2 marks each (40 Marks)

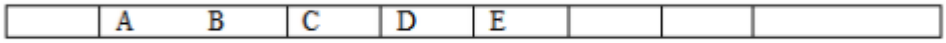
Q1.	Identify the following data structure which is an Abstract Data Type.
Option A:	Trees
Option B:	Queue
Option C:	Array
Option D:	Graph
Q2.	In linked lists, there are no NULL links in
Option A:	Circular Linked List
Option B:	Doubly Linked List
Option C:	Singly Linked List
Option D:	Multi Linked List
Q3.	Which of the following operations is performed more efficiently by doubly linked list than by singly linked list?
Option A:	Deleting a node whose location is given
Option B:	Searching of an unsorted list for a given item
Option C:	Inverting a node after the node with given location
Option D:	Traversing a list to process each node
Q4.	A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time? 
Option A:	Rear node
Option B:	Front node
Option C:	Single pointer don't support
Option D:	Node next to front

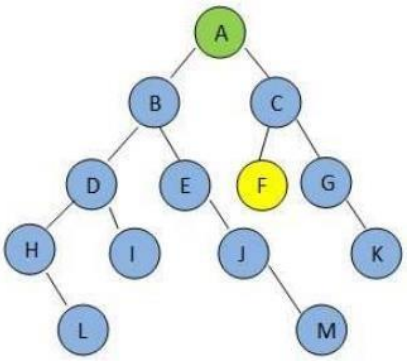
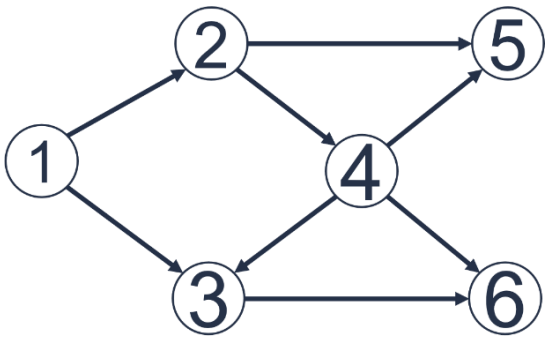
Q5.	Consider the function f defined below.
	struct item {
	int data;
	struct item * next;
	};
	int f(struct item *p) {
	return ((p == NULL) || (p ->next == NULL) ||
	((p->data <= p -> next -> data) &&
	f(p-> next)));
	}
	For a given linked list p, the function f returns 1 if and only if

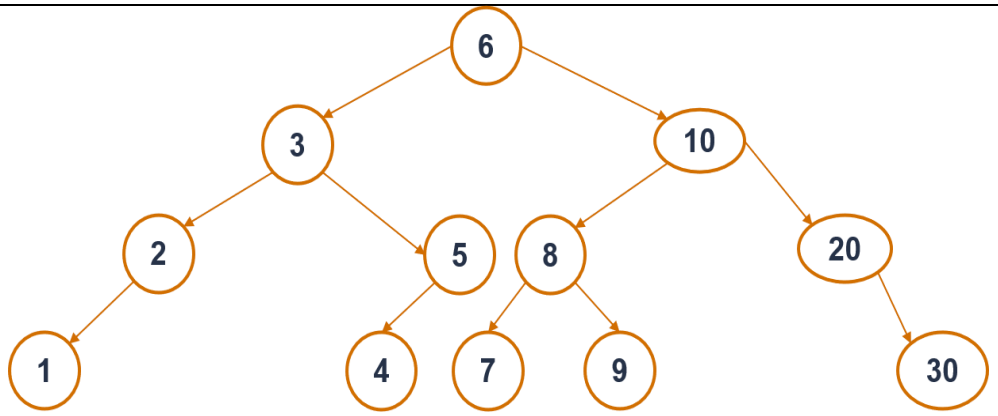
Option A:	The list is empty or has exactly one element
Option B:	The elements in the list are sorted in non-decreasing order of data valu
Option C:	The elements in the list are sorted in non-increasing order of data value
Option D:	Not all elements in the list have the same data value

Q6.	The following C function takes a simply-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank.
	
	typedef struct
	node {int value;
	struct node *next;
	} Node;
	
	Node *move_to_front(Node
	*head) {Node *p, *q;
	if ((head = = NULL: || (head->next = = NULL)) return
	head;q = NULL; p = head;
	while (p-> next
	!=NULL) {q=P;
	p=p->next;
	}
	return head;
	}
	
	Choose the correct alternative to replace the blank line.

Option A:	q = NULL; p->next = head; head = p;
Option B:	q->next = NULL; head = p; p->next = head;
Option C:	head = p; p->next = q; q->next = NULL;
Option D:	q->next = NULL; p->next = head; head = p;
Q7.	What is the outcome after the following steps, starting with an empty stack of size 5? push(3), push(5), pop(), push(10), push(11), push(100), push(9), push(10)
Option A:	Stack overflow error
Option B:	The top element is 3

Option C:	Stack underflow error
Option D:	The top element is 10
Q8.	Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence that contains 2 left parentheses and 3 right parentheses (in some order). The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation?
Option A:	1
Option B:	2
Option C:	3
Option D:	4 or more
Q9.	Consider the linear queue given below which has FRONT = 1 and REAR = 5. Now perform the following operations on the queue: (a) Add G (b) Delete two letters(c) Add H (d) Add I (e) Delete three letters 
Option A:	H,G,I
Option B:	G,H,I
Option C:	G,I,H
Option D:	H,I,G
Q10.	Which of the following is an example of stack?
Option A:	Person standing for withdrawing money
Option B:	A set of bangles worn by a lady on her arm
Option C:	Round Robin Process scheduling
Option D:	Network Printing Job
Q11.	Given a hash table of size 100, map the key 1892 to an appropriate locationin the hash table using the Multiplication function.
Option A:	30
Option B:	32
Option C:	34
Option D:	35
Q12.	A hash function h defined h(key)=key mod 7, with linear probing, is used to insert the keys 44, 79, 55, 91, 17, 63 into a table indexed from 0 to 6. What will be the location of key 17?
Option A:	3
Option B:	4
Option C:	5
Option D:	6
Q13.	Which of the following is not a limitation of binary search algorithm?
Option A:	Must use a sorted array
Option B:	Requirement of sorted array is expensive when a lot of insertion and deletions are needed
Option C:	There must be a mechanism to access middle element directly

Option D:	Binary search algorithm is not efficient when the data elements more than 1500.
Q14.	Starting from the node A at the top, which algorithm will visit the least number of nodes before visiting the node F? 
Option A:	Breadth First Search
Option B:	Depth First Search
Option C:	DFS and BFS will visit same number of nodes
Option D:	Both BFS and DFS will not visit node F
Q15.	 What will be the topological ordering for the above graph?
Option A:	1 2 3 4 5 6
Option B:	1 2 3 4 6 5
Option C:	1 3 2 4 5 6
Option D:	1 2 4 5 3 6
Q16.	To represent hierarchical relationships between elements, Which data structure is suitable?
Option A:	Stack
Option B:	Queue
Option C:	Graph
Option D:	Tree
Q17.	A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is
Option A:	$\log_2 n$
Option B:	n-1
Option C:	n/2
Option D:	n

Q18.	 What will be the Pre-order traversal output of above binary tree?
Option A:	6 3 2 1 5 4 10 8 7 9 20 30
Option B:	1 2 3 4 5 6 7 8 9 10 20 30
Option C:	1 2 4 5 3 7 9 8 30 20 10 6
Option D:	6 3 10 2 5 8 20 1 4 7 9 30
Q19.	Select the correct statement from below with respect to the M-way search tree.
Option A:	Number of Subtree may vary from 1 to M
Option B:	A node can have 1 to M-1 values in every node.
Option C:	Compulsory every node should have M-1 values
Option D:	Compulsory every node should have M subtrees.

Q20.	Consider the following code segment in C to traverse a binary search tree using the preorder
	void preorder (node *tree)
	{
	if (t)
	{
	Statement1
	Statement2
	Statement3
	}
	}
	The above Statements should be

Option A:	printf("%d", tree->info); preorder (tree->right); preorder (tree->left);
Option B:	preorder (tree->left); preorder (tree->right); printf("%d", tree->info);
Option C:	preorder (tree->left); printf("%d", tree->info); preorder (tree->right);
Option D:	printf("%d", tree->info); preorder (tree->left); preorder (tree->right);

Q2. (20 Marks)	**Solve any Four out of Six** **5 marks each**
A	Explain various operations performed on Data Structures.
B	Explain Double Ended Queue.
C	Write a function to implement following operations on doubly linked list i. Insert at end ii. Delete from beginning iii. Display
D	Construct the AVL Tree for the following numbers. 11 22 33 44 55 9 8 7 6 5
E	What are different ways to represent graph in memory?
F	Consider a hash table with size=10. Using quadratic probing insert the keys27, 72, 63, 42, 36, 18, 29 and 101 into the hash table. Take c1=1 and c2=3.

Q3. (20 Marks)	**Solve any Two Questions out of Three** **10 marks each**
A	Write a C program to implement circular linked list that performs following functions: i. Insert node in the beginning ii. Insert a node at the end iii. Display the list
B	Compute the Huffman code for each symbol.Character A B C D E F Count 9 12 5 45 16 13
C	Explain the DFS with example. Also write the program for Depth First Search.