No of Pages : 3 Course Code : 20MX13

Roll No:
(To be filled in by the candidate)

PSG COLLEGE OF TECHNOLOGY, COIMBATORE 641 004

SEMESTER EXAMINATIONS, JANUARY 2023

MCA Semester : 1

20MX13 DATA STRUCTURES

Time : 3 Hours **Maximum Marks : 100**

INSTRUCTIONS:

1. Answer **ALL** questions. Each question carries 25 Marks.
2. Course Outcome : Table

Qn.1	CO 1	Qn.2	CO 2	Qn.3	CO 3	Qn.4	CO 4

1. a) What is Abstract data type (ADT)? Define String ADT. (5)

   b) How are multidimensional matrices stored in memory? Write the addressing function corresponding to each representation. A Toeplitz Matrix is a square matrix in which the elements on a diagonal are same. Following is an example of Toeplitz Matrix:

10	7	3	4	5
6	10	2	3	4
7	6	10	2	3
8	7	6	10	2
9	8	7	6	10

   Suggest an efficient storage representation for storing Toeplitz Matrix and write the corresponding addressing function. (8)

   c) Formulate algorithms for linear and binary search. Trace the algorithms on sample input to show the correctness of the algorithm. Write the best and worst case complexity of both algorithms. (12)

2. a) What is a circular Queue? Highlight the difference between linear and circular queue. Write algorithms for the primitive operations on circular queue. (5)

   b) Write an algorithm to check the well formedness of parenthesis in an expression. Consider (,), [,], { and } types of parentheses. Trace the algorithm on the inputs {a + [((c-d) * e)] } and {a + [c-(a+d])}. (8)

   c) i) Write algorithms for the primitive operations on array based stack. Formulate an algorithm converting an infix expression to postfix form using stacks. Trace the algorithm on the expression a * (b + c *d) / e. (12)

   (OR)

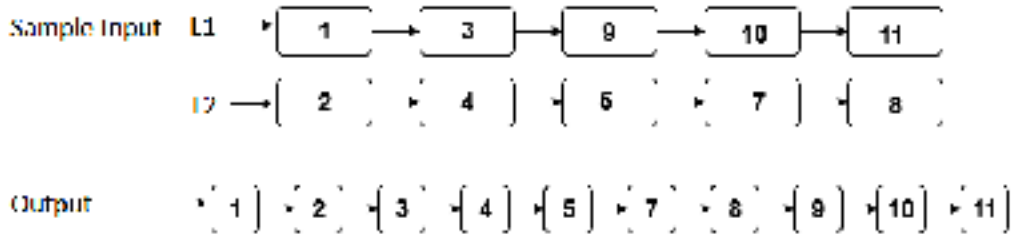
   ii) a. What is a dequeue? What are the different types of dequeue? (4)

   b. Formulate an algorithm to evaluate postfix expressions. Trace the algorithm on the expression abc*+cd-* where a = 2, b = 4, c = 2, d = 1. (8)

3. a) i. Differentiate static and dynamic memory allocation. (2)

   ii. Write an algorithm to count the number of nodes in a circularly linked list. (3)

b) Formulate an algorithm to merge two sorted lists. Sample input and corresponding output are shown below: (8)



c) i) 1) How can a polynomial be represented using linked lists? Represent $5x^5 + 4x^3 + 3$ using linked lists. Write an algorithm to add two polynomials represented using linked lists. (8)

2) Represent the following sparse matrix as a linked list. (4)

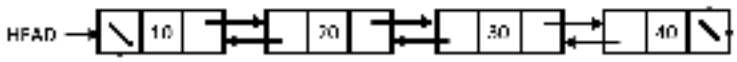
0	0	3	0
0	0	0	5
1	0	3	0
0	0	7	0

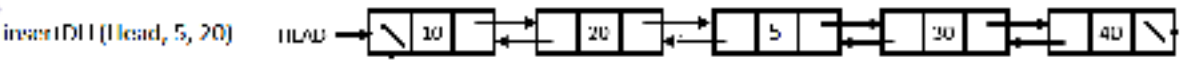
(OR)

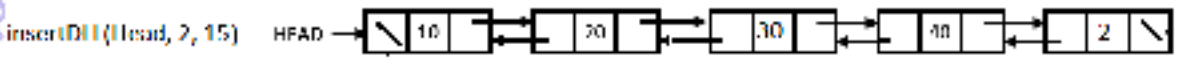
ii) 1) What is a linked queue? Write algorithms for the primitive operations on linked queue. (6)

2) Write an algorithm insertDLL(Head, elt, value) to add elt, in a doubly linked list next to the given value if the value is present in the list. Otherwise add the element at the end of the list.

Sample inputs and outputs are shown below: (6)







4. a) Write the order in which the nodes are visited in the different types of traversals in a Binary tree. Draw the binary tree whose inorder traversal sequence is: g d h b e i a f j c and preorder traversal sequence is: a b d g h e i c f j. (5)

b) i) What are the different types of representation of graphs? Give an example for the following types of graphs and represent them using various types of representations :

Directed Graph, Undirected weighted graph. (4)

ii) Traverse the graph shown in Fig 1 using Breadth First Search (BFS) and Depth First Search (DFS) and write the traversal sequences. Indicate the data structure used in BFS and DFS. (4)

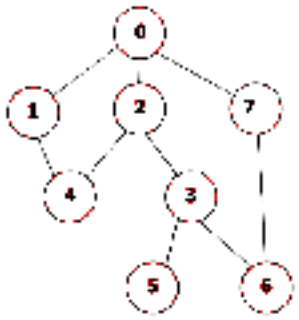


Fig 1

c) What is hashing? Give any two hash functions. What are collisions in hashing? What are the ways of resolving collisions? Describe how elements are located with different types of collision resolution methods. How can the hash performance be improved? Consider the following items to be inserted into a hash table : 69, 66, 80, 35, 18, 60, 89, 70, 12 and a hash table of size 11. Use any one hash function of your choice and illustrate all types of collision resolution techniques. (12)

/END/

FD/RL