Group – D

6. (a) Given a binary tree whose inorder and preorder traversal are given by
Inorder: E I C F B G D J H K
Preorder: B C E I F D G H J K
Draw the tree. What will be the post order traversal of the binary tree?

(b) Insert the following entries, in the order stated, into an initially empty B-tree of order 5,
A G F B K D H M J E S I R X C L N T U P

(3 + 1) + 8 = 12

7. (a) Insert the keys, in the order shown, to build them into an AVL tree. Clearly show the rotation(s) when required.
A, V, L, T, R, E, I, S, O, K, F, J

(b) Draw a Trie constructed from the words
a ear re rare area are ere era rarer
built from the letters a, e, r.

8 + 4 = 12

Group – E

8. (a) Describe the general idea of linear search. How is the idea of sentinel used to increase the efficiency of linear search?

(b) Apply binary search to search for the element 91 in the following list:
91 88 81 73 62 30 13

(c) Trace the action of radix sort on the following list of seven numbers considered as three-digit integers: 265, 337, 357, 295, 193, 125, 224.

(3 + 2) + 3 + 4 = 12

9. (a) Write an algorithm for quicksort clearly mentioning your choice for pivot element. Find an ordering of the keys that will force your algorithm into its worst case.

(b) What is hashing? A hash function f is defined as f (key) = key mod 13, with linear probing to resolve collision. Insert the keys 55, 58, 68, 91, 27, 145, 79 into the table indexed from 0 to 12.

(6 + 1) + (2 + 3) = 12

DATA STRUCTURES
(MCAP 1202)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) Given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?
(a) Delete the first element
(b) Insert a new element as a first element
(c) Delete the last element
(d) Insert a new element at the end of the list.

(ii) Consider the following declaration for a two-dimensional array in C:
char a[100][100];
Assuming that the main memory is byte addressable and that the array is stored starting from memory address 0, the address of a[40][50] is
(a) 4040 (b) 4050 (c) 5040 (d) 5050.

(iii) What is the most appropriate matching for the following pairs?

X. m = malloc (5); m = NULL;	1. Using dangling pointers
Y. free (n); n→value = 5;	2. Using uninitialized pointers
Z. char *p; *p = 'a';	3. Lost memory

(a) X-1,Y–3, Z–2 (b) X–2,Y–1, Z–3
(c) X–3,Y–2, Z–1 (d) X–3,Y–1, Z–2.

(iv) The following sequence of operations is performed on a stack:
push (10), push (20), pop, push (10), push (20), pop, pop, pop, push (20), pop.
The sequence of values popped out is
(a) 20, 10, 20, 10, 20 (b) 20, 20, 10, 10, 20
(c) 10, 20, 20, 10, 20 (d) 20, 20, 10, 20, 10.

(v) What is the value of the postfix expression: 8 3 4 + – 3 8 2/+ * 2 ^ 4 + (Considering ^ is used for exponentiation)
(a) 17 (b) 131 (c) 64 (d) 53.

(vi) Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are
(a) full: REAR = = FRONT; empty: (REAR + 1) % n = = FRONT
(b) full: (REAR + 1) % n = = FRONT; empty: (FRONT + 1) % n = = REAR
(c) full: (REAR + 1) % n = = FRONT; empty: REAR = = FRONT
(d) full: (FRONT + 1) % n = = REAR; empty: REAR = = FRONT.

(vii) A binary tree with 27 nodes has _____ null branches.
(a) 54 (b) 27 (c) 26 (d) 28.

(viii) Which of the following statements is true for the function $T(n) = 3T(n/2) + n^2$?
(a) $T(n) = \Theta(n^2)$ (b) $T(n) = \Theta(n^2 \log n)$
(c) $T(n) = \Theta(n \log n)$ (d) $T(n) = \Theta(n)$.

(ix) Number of binary trees formed with 4 nodes is
(a) 16 (b) 4 (c) 8 (d) 14.

(x) Given two sorted list of size 'm' and 'n' respectively. The number of comparisons needed in the worst case by the merge sort algorithm to merge them into a single sorted list will be
(a) m × n (b) maximum of m and n
(c) m + n – 1 (d) minimum of m and n.

Group – B

2. (a) Define Big-Oh notation in relation to growth rate of a function.

(b) Determine whether the following statement is correct
$2n^2 + n\log n = \Theta(n^2)$

(c) Compare between linear and non-linear data structure.

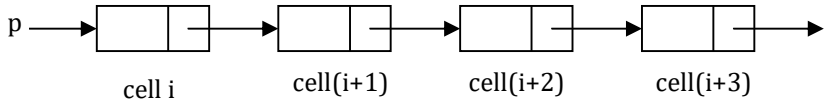
(d) What is a sparse matrix? Implement a sparse matrix by linked list.

2 + 3 + 2 + (1 + 4) = 12

3. (a) A node of a singly liked list is declared as

```
typedef struct node
{
    int data;
    struct node *next;
} NODEPTR;
```

Let p be a pointer as shown in the figure to the singly linked list:



Let q be a pointer to NODEPTR
The statements below are executed sequentially. Illustrate what happens after execution of each statement.
q = p → next;
p → next = q → next;
p → next = (q → next) → next;
(p → next) → next = q;

(b) Compare the relative advantages and disadvantages between an array and a linked list.

(c) Write an algorithm to reverse a singly linked list, traversing the list exactly once.

4 + 4 + 4 = 12

Group – C

4. (a) State the Towers of Hanoi problem. Devise a solution to the Towers of Hanoi problem, clearly explaining the precondition and postcondition for your algorithm. Derive the time complexity of your algorithm in terms of Big-Oh notation.

(b) How does the time requirement and space requirement of a recursive function relate to its recursion tree?

(2 + 5 + 3) + (1 + 1) = 12

5. (a) What is a circular queue? What are the advantages of circular queue over a simple linear queue?

(b) Are recursive routines more efficient than non-recursive ones? Justify your answer.

(c) Suppose 'list' is the external pointer to a singly linked list. A node of the linked list contains two fields, an integer 'info', and a pointer to node 'next'. Explain the output when the following function is called as fun(list)? Will the function act differently depending on whether the number of nodes in the list is even or odd?

```
void fun(struct node *p){
    if(p == NULL)
            return;
    printf("%d  ", p->info);
    if(p->next != NULL )
            fun(p->next->next);
    printf("%d  ", p->info);
}
```

(1 + 3) + (1 + 3) + (3 + 1) = 12