| Reg. No. | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

# Manipal Institute of Technology, Manipal

(A Constituent Institute of Manipal University)

## III SEMESTER B.TECH (COMPUTER SCIENCE AND ENGINEERING)
## END SEMESTER EXAMINATIONS, NOV/DEC 2015

SUBJECT: **DATA STRUCTURES (CSE 2103)**
**REVISED CREDIT SYSTEM**
DATE: 01-12-2015

Time: 3 Hours MAX. MARKS: 50

**Instructions to Candidates:**

- Answer **ALL** the questions.
- Missing data, if any, may be suitably assumed.

1A. The Fibonacci sequence 0, 1, 1, 2, 3, 5, … is defined as F0 = 0, F1 = 1 and Fi = F(i-1) + F(i-2), i >= 2. Write a recursive function fib(n), that returns the nth Fibonacci number. Show the call tree for the call fib(5). What can you say about the efficiency of this solution in comparison to that of iterative solution? Under what conditions recursion is then ideal? 3M

1B. Write a binary search function using pointers. Name the function as *int binarysearch(int list[], int *endptr, int key, int **locn)* which will search for a *key* in the given *list* and *endptr* is pointer to the largest element in the list, *locn* is pointer to the key if the key is found. The function should return 0 if the key is not found. 4M

1C. Write a program that will construct a ragged array namely *table* by asking the user for row and column size, namely *r* and *c*. The program will then construct the table dynamically of size *[r, c+1]* and stores the entered values into all of the *c* locations while the first location in all the rows is held by the average of all the values present in the given row. Write the entire main() code without using any function calls. Include the code for displaying the content of entire ragged array. 3M

2A. Suppose a circular queue is maintained as an array with n= 8 elements in which front and rear are initialized to zero. Find the size of the left segment, size of the right segment and the total number of elements in the queue for the following cases. In each case, show the queue configuration clearly marking front and rear in the queue array and show the array contents by character data.

i) front = 5, rear =3
ii) front = 7, rear =5
iii) front = 3, rear =1 3M

2B. For a circular queue of size 8 maintained as an array, how many configurations exist for a full circular queue? Show all of them. Among them, how many are for 2-segments and 1-segments? In each configuration, clearly mark front and rear and show the array contents by character data. Initially, the front and rear index of a circular queue are reset to zero. 3M

2C. Write an algorithm for converting an infix expression to a postfix form. Use this algorithm to trace the infix expression (( P – ( Q + R)) * S) $ ( T + U) in the form of a table displaying symbol, postfix string and stack contents from left to right order. What content is finally remained in the stack on encountering end of input which is then attached to the postfix string? 4M

3A. Discuss the advantages, if any, of deleting a node whose data matches the given key from a doubly linked list in comparison to that of a singly linked list. Illustrate the advantages by writing a function namely *struct node *deleteDLL(struct node*head, key)* and show that in comparison to

singly linked list namely *struct node *deleteSLL(struct node *head, key).* Assume that the linked lists are available with a pointer named head pointing to the header node. Also assume that a node of singly linked list contains a pointer 'next' to the next node whereas the node of a doubly linked list contains two pointers 'prev' and 'next' to point to the previous node and the next node respectively. 5M

3B. Write a program that reads a [3*3] matrix M and creates three circular singly linked lists namely *list1, list2* and *list3* which store first, second and third row of the matrix M.

i) The function *struct node *add ( struct node *list, int item)* to attach nodes to the circular singly linked lists

ii) Next, the function *struct node*findRowSum(struct node *list)* to compute row sum by accessing the linked lists. After finding the row sum of a list, each list is to be appended at the end with a node that holds the row sum.

iii) Similarly, the fourth linked list namely list4 is to be created to store column sum from three linked lists which will access all ith elements of each list to compute column sum and appends as ith node of list4. The function named *struct node *findColSum(struct node *list1, struct node *list2, struct node *list3, struct node *list4)* is to be written.

*iv)* Finally a function named *struct node *appendMat(struct node *list1, struct node *list2, struct node *list3, struct node *list4)* that will append the contents of four lists into 4 rows of a [4*4] matrix using *for* loop.

v) Write a main() function to access all the above functions. 5M

4A. Considering the level order numbering scheme of a complete binary tree in which the root node is numbered 1 and then onwards for each level, the nodes are numbered sequentially from left to right, arrive at an expression for finding the distance of a node from the root in terms of its node number. The distance of a node from the root should find the number of edges between the root and the node. Use this expression to compute the distance of a node at the $5^{th}$ level of a zig-zag binary tree in which there is only a left child from the root and then a right child at the next level and then similarly the left child at one level and the right child at the next level gets continued till the last level. Assume the level of a root node is zero. Compute the internal path length of the above tree. 3M

4B. Write an iterative function named *struct tree* add (struct tree *root, struct employee emp)* that inserts the employee details ordered by the employee code into a binary search tree. Assume that an employee record contains name, empcode and salary as its fields. 4M

4C. For the elements < F, G, B, D, A, C, E, I, H>, draw the binary search tree assuming F as the root and inserting elements in the order specified. Transform the binary search tree into a threaded binary tree and draw the threaded binary tree representation with header node. Write the code for finding inorder successor using function *struct tree* findSucc(struct tree *root).* Here *root* points to the header node of the tree. For the given elements, trace the function for the threaded tree and display the result in the table form showing node and its inorder successor. In the table, show all the nodes starting from the header node until you reach the header node in such a way that the successor generated is to be provided as next input to the function. 3M

5A. Stating the property of red black trees, show why it is not possible to construct a chain of 3 nodes as a red-black tree. Illustrate with different combinations of red-black trees. 2M

5B. Distinguish between the following:

i) M-way search tree and B-trees

ii) Adjacency matrix and adjacency list representation of undirected graphs 4M

5C. Determine the following:

i) Iteration-wise output from insertion sort for a list of 5 numbers <50, 40, 30, 20, 10> to sort in ascending order

ii) Cost of a Binary search tree for elements <10, 20, 30, 40, 50, 60, 70> arranged in a complete binary tree form assuming equal probabilities for internal and external nodes. 4M

**********