(28)

[This question paper contains 8 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 4930 **G**

Unique Paper Code : 42347501

Name of the Paper : Data Structures

Name of the Course : **B.Sc. Programme (CBCS-LOCF)**

Semester : V

Duration : 3 Hours Maximum Marks : 75

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **Section A** is Compulsory.
3. Attempt any **Five** Questions from **Section B**.
4. All parts of a question must be answered together.

P.T.O.

PART A

(**All** *questions in this section are compulsory.*)

1. (a) Add the two integers 8674 and 321 usi stacks. Show the contents of the stacks at ea step. (

(b) Which data structure is best suited to print a s of elements in reverse order and why? Expla using your own example. (

(c) Consider the following function : (

```
void fun(int n) {
    queue<int> Q;
    Q.enqueue(0); Q. enqueue(1);
    for (int i = 0; i < n; i++ ) {
        int a = Q.dequeue();
        int b = Q.dequeue();
```

```
        cout << b << " ";
        Q.enqueue (a+b); Q.enqueue(b);
    }
}
```

What is the output if the function is invoked as fun(4)? Show the contents of the queue at each iteration.

(d) Write a function to remove an element from the i^{th} position in a singly linked list. (4)

(e) Write a recursive function for Linear Search on an array of integers. The function should return the index of the element if it is found else it should return -1. (4)

(f) Assume a binary tree node contains fields for data, a pointer to the left child(left), and a pointer to the right child(right). Determine the functionality

of the following function if it is invoked with th root of a binary tree.

```
int fun(Treenode* p) {
    if (p == 0) return 0;
    return fun(p->left) + fun(p->right) + 1;
}
```

What will the function return if the root points a full and complete binary tree of height 3?

PART B

(Attempt any five)

2. (a) Write a function to merge two sorted arrays in a new sorted array.

(b) Convert the following infix expression to pref expression. Show the status of the stack at ea step :

Infix expression: 5 + 6 / 3 * 2 + 7

3. (a) Write a recursive algorithm for implementing binary search. Show steps involved in searching an element with value 42 in the following array of integers :

{5, 16, 23, 32, 37, 42, 55} (5)

(b) Perform selection sort to sort the following list of integers, show all the steps performed in detail : {53, 32, 96, 64, 22, 11}. (5)

4. (a) Write the enqueue and dequeue functions of a queue implemented using a singly linked list. (5)

(b) What is a priority queue and why is it required? Predict the output of the following code which uses two priority queues. (5)

```
int main() {

    int a[]= {1,2,3,4,5,6,7,8,9,10};

    priority_queue<int,vector<int>> pq1(a+3,a+7);
```

P.T.O.

```
priority_queue<int,vector<int>,greater<int
pq2(a+3,a+7);

while (!pq1.empty()) {

    cout << pq1.top() << ' '; pq1.pop();

}

cout << endl;

while (!pq2.empty()) {

    cout << pq2.top() << ' '; pq2.pop();

}

}
```


5. (a) Give necessary class definitions to create a dou linked list. Write a member function to delete element from the i^{th} position in a doubly lin list.

(b) Write a function to search an element 'x' in a singly linked list of integers. The function should return true if the element is found else return false. (5)

6. (a) Consider the following recursive function : (5)

```
double compute(double a, int n) {
    if (n == 0)
        return 1.0;
    else
        return a * compute(a, n-1);
}
```

How many recursive calls will be performed to obtain the result of compute(3,2)? Show the changes in the run time stack during execution of compute(3,2).

(b) Write a recursive function to calculate sum of the digits of a given number. (5)

7. (a) Write a program for level by level traversal of a binary tree. (5)

(b) What is the minimum possible and maximum possible number of nodes in a binary tree with height 'h'? Given h = 4, draw both trees – one with minimum number and one with maximum possible number of nodes. (5)