



# BMSCOLLEGE OF ENGINEERING

(Autonomous Institute, Affiliated to VTU)

**Department of Information Science and Engineering**

## INTERNALS - II

**Course Code: 19IS3PCDSC**

**Course Title: Data Structures with C**

**Semester: III**

**Maximum Marks: 40**

**Date: 1<sup>st</sup> December 2020**

**Faculty Handling the Course:**

**Dr.HSG, Dr. BRS, Prof. RIS**

**Instructions: Internal choice is provided in Part C.**

### PART-A

**Total 5 Marks(No choice, each question should be answered)**

No.	Question	Marks
1	<p>Sketch the memory representation (sequential and linked) for the following Binary tree.</p> <pre> graph TD     A((A)) --&gt; B((B))     A --&gt; C((C))     B --&gt; D((D))     B --&gt; E((E))     C --&gt; F((F))     C --&gt; G((G))     E --&gt; H((H))     E --&gt; I((I))           </pre>	5

### PART-B

**Total 15 Marks (No Choice, each question should be answered)**

No.	Question	Marks
2. a.	<p>Write the inorder, preorder and postorder traversals for the following Binary tree.</p> <pre> graph TD     L1((L)) --&gt; M((M))     L1 --&gt; Q((Q))     M --&gt; T((T))     M --&gt; O((O))     Q --&gt; P((P))     Q --&gt; R((R))     O --&gt; N((N))     O --&gt; S((S))     P --&gt; L2((L))           </pre>	5

2. b.	<p>Analyze the given recursive routine below. Describe the logic and predict the return value by assuming game[ ] array of at least 5 elements of your own.</p> <pre> int hockey(int game[], int begin,int end) {     if(begin == end)         return game[begin];     int mid = (begin + end)/2;     int x = cricket(game, begin, mid);     int y = cricket(game, mid+1, end);     if(x &lt; y)         return x;     else         return y; } </pre>	5
2. c.	<p>Complete the following C routines which implement Linear Queue operations using arrays</p>	5

```

#include<stdio.h>
#define MAX 10;
struct queue
{
    int front;
    int rear;
    int q[MAX];
};
struct queue metro;
metro.front = ____;
metro.rear = ____;
void insertLQ(int item)
{
    if(_____)
        printf("Queue is FULL\n");
    else
        metro.q[metro.rear] = item;
        _____;
}

void deleteLQ()
{
    if(metro.front == metro.rear)
        printf("No Elements\n");
    else
        printf("The deleted element is %d", _____);
        _____;
}

```

### PART- C

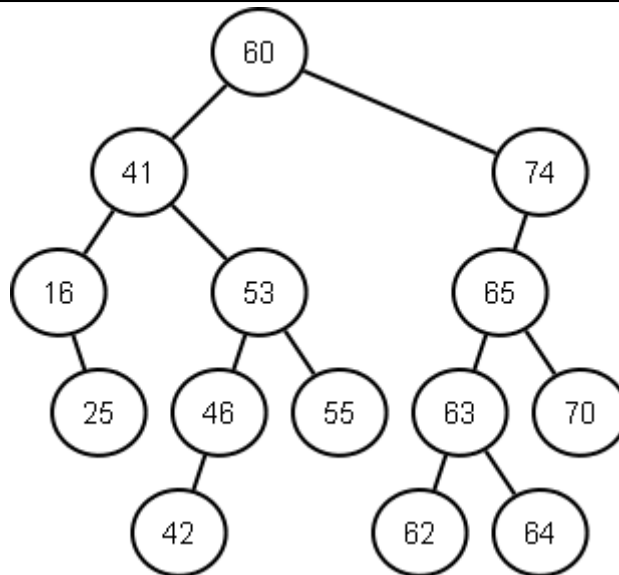
**Total 20 Marks**

**(Answer any one question from 3a, 3b**

**and**

**Answer any one question from 3c, 3d)**

No.	Question	Marks
3.a.	Show how nodes <b>25</b> , <b>46</b> and <b>65</b> are deleted successively from the following Binary Search Tree. Also write the algorithm.	10



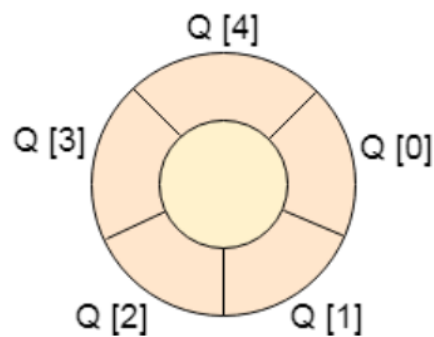
Or

3. b. Consider an empty queue with maximum size 5. Determine the values of front and rear when the following statements are executed sequentially. Sketch the circular queue (as shown in the figure below) after each statement is executed. Also write C program for the implementation of circular queue.

10

```

insert(11);
insert(9);
insert(26);
insert(40);
delete();
insert(56);
delete();
insert(32);
insert(15);
delete();
  
```



3. c.	Write C routines to implement the following for a Binary Search Tree: i. Count the numbers of nodes with exactly one child. ii. Identify and print minimum value. iii. Determine the height of the tree.	10
<b>Or</b>		
3. d.	Trace and write the algorithms for the following: i. Tower of Hanoi with 3 discs ii. Find GCD of two integers for $n1 = 242$ and $n2 = 330$	10