OR	iv.	Write a c program to check weather a given matrix is upper triangular or not.	**5**
Q.4	i.	Distinguish between stack and queue. Explain in detail four differences with example.	**4**
	ii.	Convert the following infix expression to postfix notation and prefix notation using stack. A+(B*C-(D/E^F)*G)*H	**6**
OR	iii.	Write an algorithm to convert infix expression to postfix form using stack.	**6**
Q.5		Attempt any two	
	i.	What is AVL Tree? Explain right and left rotations with the help of an example.	**5**
	ii.	Represent the following expression using a tree. (a-b) / ((c*d)+e) What you get when this tree is traversed in Preorder, and postorder.	**5**
	iii.	What is linked list? Explain with example. Write a c function which inserts a node at the last position.	**5**
Q.6	i.	Which sorting algorithm is best if the list is already sorted? Why?	**2**
	ii.	Write the steps to sort following list 8, 22, 7, 9, 31, 5, 13 in ascending order, using bubble sort.	**3**
	iii.	What is hashing technique? State the advantages and disadvantages of hashing technique.	**5**
OR	iv.	What is Graph? Explain Depth First Search traversal of Graph using an example	**5**

******

Total No. of Questions: 6 *Total No. of Printed Pages:4*

Enrollment No.....

Faculty of Science

End Sem (Even) Examination May-2019

CA3CO07 Data Structure

Programme: BCA Branch/Specialisation: Computer Application

Duration: 3 Hrs. **Maximum Marks: 60**

Note: All questions are compulsory. Internal choices, if any, are indicated. Answers of Q.1 (MCQs) should be written in full instead of only a, b, c or d.

Q.1 i. How many times is the comparison i <= j performed in the following program? **1**

```
int i = 110, j = 200;
void main()
{
  while (i <= j){
    i = i+1;
    j= j-1;
  }
}
```

(a) 46 (b) 47 (c) 45 (d) 48

ii. If for an algorithm time complexity is given by $O(3^n)$then complexity will: **1**

(a) Constant (b) Quadratic

(c) Exponential (d) None of these

iii. consider the size of integer 2 bytes **1**

```
# include <stdio.h>
 void display(int array[])
{
  int n = sizeof(array[1]);
  int i;
  for (i = 0; i < n; i++)
    printf("%d  ", array[i]);
}
```

P.T.O.

```
int main()
{
  int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
  display(arr);
  return 0;
}
```

(a) 1 (b) 1 2 (c) 1 1 (d) 2 1

iv. Which of the following is true about arrays in C? **1**

(a) For every type T, there can be an array of T.
(b) For every type T except void and function type, there can be an array of T.
(c) When an array is passed to a function, C compiler creates a copy of array.
(d) 2D arrays are stored in column major form

v. How many stacks are needed to implement a queue? Consider the situation where no other data structure like arrays, linked list is available to you. **1**

(a) 1 (b) 2 (c) 3 (d) 4

vi. A normal queue, if implemented using an array of size MAX_SIZE, gets full when **1**

(a) Rear = MAX_SIZE – 1
(b) Front = (rear + 1)mod MAX_SIZE
(c) Front = rear + 1
(d) Rear = front

vii. The following C function takes an input from user and adds it in simply-linked list as first node. Some part of the code is left blank. Choose the correct alternative to replace the blank line. **1**

```
typedef struct NODE
{
 int value;
 struct node *next;
}node;
 node *head=NULL
```



```
void insert_to_front(int v)
{
 node *p;
 p = (node *)malloc(sizeof(node);
 p->value=v;
 if(head==NULL)
 {  _________  }
else
  { p->next=head;}
 head=p;
}
```

(a) p->next=p (b) p->next=head
(c) p->next=NULL (d) p->next=&v

viii. What is the maximum height of an AVL tree with n nodes? **1**

(a) n (b) log(n) (c) (n*n)+1 (d) n⁄2

ix. What is the worst case complexity of bubble sort? **1**

(a) O(nlogn) (b) O(logn) (c) O(n) (d) O(n^2)

x. What can be the techniques to avoid collision? **1**

(a) Use binary search (b) Use the chaining method
(c) Use linear search (d) All of these

Q.2 i. Explain the term data and data type. **2**

ii. Define the Order of growth of an algorithm. **3**

iii. What is data structure? Explain various types of data structure in detail. **5**

OR iv. List out the areas in which data structures are applied Extensively? Explain five areas in detail. **5**

Q.3 i. What is ordered List? **2**

ii. Describe the storage structure of 2D Array. **3**

iii. Explain strcmp and strcat functions defined in string.h with example. **5**

Marking Scheme
CA3CO07 Data Structure

Q.1	i.	How many times is the comparison i <= j performed in the following program? (b) 47		**1**
	ii.	If for an algorithm time complexity is given by $O(3^n)$then complexity will: (c) Exponential		**1**
	iii.	consider the size of integer 2 bytes (b) 1 2		**1**
	iv.	Which of the following is true about arrays in C? (b) For every type T except void and function type, there can be an array of T.		**1**
	v.	How many stacks are needed to implement a queue? Consider the situation where no other data structure like arrays, linked list is available to you. (b) 2		**1**
	vi.	A normal queue, if implemented using an array of size MAX_SIZE, gets full when (a) Rear = MAX_SIZE – 1		**1**
	vii.	The following C function takes an input from user and adds it in simply-linked list as first node. Some part of the code is left blank. Choose the correct alternative to replace the blank line. (c) p->next=NULL		**1**
	viii.	What is the maximum height of an AVL tree with n nodes? (b) log(n)		**1**
	ix.	What is the worst case complexity of bubble sort? (d) $O(n\text{^}2)$		**1**
	x.	What can be the techniques to avoid collision? (b) Use the chaining method		**1**
Q.2	i.	Data definition Data type.	1 mark 1 mark	**2**
	ii.	Order of growth of an algorithm.		**3**
	iii.	Data structure definition Types of data structure	2 marks 3 marks	**5**
OR	iv.	Areas in which data structures are applied Extensively 1 mark for each application	(1 mark * 5)	**5**
Q.3	i.	Ordered List definition		**2**
	ii.	Storage structure of 2D Array.		**3**
	iii.	Strcmp() strcat ()	2.5 marks 2.5 marks	**5**
OR	iv.	c program to check weather a given matrix is upper triangular or not.		**5**
Q.4	i.	Distinguish between stack and queue. 1 mark for each difference	 (1 mark * 4)	**4**
	ii.	Convert the following infix expression to Postfix notation Prefix notation using stack.	 3 marks 3 marks	**6**
OR	iii.	Algorithm to convert infix expression to postfix form using stack.		**6**
Q.5		Attempt any two		
	i.	AVL Tree & right and left rotations with example.		**5**
	ii.	Tree diagram Preorder traversal Postorder traversal	2 marks 1.5 marks 1.5 marks	**5**
	iii.	Linked list definition Example C function	1 mark 1 mark 3 marks	**5**
Q.6	i.	Sorting algorithm name Reason	1 mark 1 mark	**2**
	ii.	Write the steps to sort following list Step by step solution		**3**
	iii.	Hashing technique Advantages Disadvantages of hashing technique	2 marks 1.5 marks 1.5 marks	**5**
OR	iv.	Graph definition Depth First Search Example	2 marks 2 marks 1 mark	**5**

******