



**Maratha Vidya Prasarak Samaj's**  
**Rajarshi Shahu Maharaj Polytechnic, Nashik**

**Udoji Maratha Boarding Campus, Near Pumping Station, Gangapur Road, Nashik-13.**

**Affiliated to MSBTE Mumbai, Approved by AICTE New Delhi, DTE Mumbai & Govt. of Maharashtra, Mumbai.**

---

*Subject: - Data Structure Using 'C'*  
*(22317)*



# SYLLABUS

<b>Chapter No.</b>	<b>Name of chapter</b>	<b>Marks Without Options</b>	<b>Marks With Options</b>
<b>1</b>	Introduction To Data Structures	06	08
<b>2</b>	Searching And Sorting	12	18
<b>3</b>	Stacks And Queues	20	30
<b>4</b>	Linked Lists	16	24
<b>5</b>	Trees And Graphs	16	22
<b>Total Marks :-</b>		<b>70</b>	<b>102</b>



## BOARD THEORY PAPER PATTERN FOR DSU[22317]

Q.1	Attempt any FIVE	5*2=10	Course Outcome (CO)
	a) Introduction To Data Structures		CO-317.1
	b) Stacks And Queues		CO-317.3
	c) Trees And Graphs		CO-317.5
	d) Stacks And Queues		CO-317.3
	e) Trees And Graphs		CO-317.5
	f) Introduction To Data Structures		CO-317.1
	g) Stacks And Queues		CO-317.3
Q.2	Attempt any THREE	3*4=12	
	a) Searching And Sorting		CO-317.2
	b) Linked Lists		CO-317.4
	c) Stacks And Queues		CO-317.3
	d) Trees And Graphs		CO-317.5
Q.3	Attempt any THREE	3*4=12	
	a) Introduction To Data Structures		CO-317.1
	b) Stacks And Queues		CO-317.3
	c) Searching And Sorting		CO-317.2
	d) Trees And Graphs		CO-317.5
Q.4	Attempt any FOUR	3*4=12	
	a) Searching And Sorting		CO-317.2
	b) Trees And Graphs		CO-317.5
	c) Linked Lists		CO-317.4



**Maratha Vidya Prasarak Samaj's**  
**Rajarshi Shahu Maharaj Polytechnic, Nashik**

Udoji Maratha Boarding Campus, Near Pumping Station, Gangapur Road, Nashik-13.

Affiliated to MSBTE Mumbai, Approved by AICTE New Delhi, DTE Mumbai & Govt. of Maharashtra, Mumbai.

	d)	Stacks And Queues	CO-317.3
	e)	Linked Lists	CO-317.4
<b>Q.5</b>		<b>Attempt any TWO</b>	<b>2*6=12</b>
	a)	Stacks And Queues	CO-317.3
	b)	Trees And Graphs	CO-317.5
	c)	Linked Lists	CO-317.4
<b>Q.6</b>		<b>Attempt any Two</b>	<b>2*6=12</b>
	a)	Searching And Sorting	CO-317.2
	b)	Stacks And Queues	CO-317.3
	c)	Linked Lists	CO-317.4



# **CLASS TEST - I**

## **PAPER PATTERN**

### **DSU[22317]**

#### **Syllabus:-**

<b>Unit No.</b>	<b>Name of the Unit</b>	<b>Course Outcome (CO)</b>
<b>1</b>	Introduction To Data Structures	CO-317.1
<b>2</b>	Searching And Sorting	CO-317.2
<b>3</b>	Stacks	CO-317.3

<b>Q.1</b>	<b>Attempt any FOUR</b>	<b>4*2=8Marks</b>	<b>Course Outcome (CO)</b>
<b>a)</b>	Introduction To Data Structures		CO-317.1
<b>b)</b>	Searching And Sorting		CO-317.2
<b>c)</b>	Stacks		CO-317.3
<b>d)</b>	Searching And Sorting		CO-317.2
<b>e)</b>	Introduction To Data Structures		CO-317.1
<b>f)</b>	Stacks		CO-317.3
<b>Q.2</b>	<b>Attempt any THREE</b>	<b>3*4=12 Marks</b>	
<b>a)</b>	Searching And Sorting		CO-317.2
<b>b)</b>	Introduction To Data Structures		CO-317.1
<b>c)</b>	Searching And Sorting		CO-317.2
<b>d)</b>	Stacks		CO-317.3



Maratha Vidya Prasarak Samaj's

**Rajarshi Shahu Maharaj Polytechnic, Nashik**

Udoji Maratha Boarding Campus, Near Pumping Station, Gangapur Road, Nashik-13.

Affiliated to MSBTE Mumbai, Approved by AICTE New Delhi, DTE Mumbai & Govt. of Maharashtra, Mumbai.

# CLASS TEST - II

## PAPER PATTERN

### DSU[22317]

Syllabus:-

Unit No.	Name of the Unit	Course Outcome (CO)
1	Queues	CO-317.3
4	Linked Lists	CO-317.4
5	Trees And Graphs	CO-317.5

Q.1	Attempt any FOUR	4*2=8Marks	Course Outcome (CO)
a)	Queues		CO-317.3
b)	Queues		CO-317.3
c)	Linked Lists		CO-317.4
d)	Linked Lists		CO-317.4
e)	Trees And Graphs		CO-317.5
f)	Trees And Graphs		CO-317.5
Q.2	Attempt any THREE	3*4=12 Marks	
a)	Queues		CO-317.3
b)	Queues		CO-317.3
c)	Linked Lists		CO-317.4
d)	Trees And Graphs		CO-317.5



# COURSE OUTCOME (CO)

**COURSE:- DSU (22317)**

**PROGRAMME: - CM**

<b>CO.NO</b>	<b>Course Outcome</b>
CO-317.1	Perform basic operations on arrays.
CO-317.2	Apply different searching and sorting techniques.
CO-317.3	Implement basic operations on stack and queue using array representation.
CO-317.4	Implement basic operations on Linked List.
CO-317.5	Implement program to create and traverse tree to solve problems.



# 1. Introduction To Data Structures

Position in Question Paper

Total Marks-06

Q.1. a) 2-Marks.

Q.1. f) 2-Marks.

Q.3. a) 4-Marks.

## Descriptive Questions

1. Define the term algorithm.
2. Differentiate between linear and non-linear data structures on any two parameters.
3. List any four operations on data structure.
4. Define Abstract Data Type.
5. Write any four applications of data structure.
6. Explain the term: i) Time complexity ii) Space Complexity.
7. Write 'C' program for deletion of an element from an array.
8. Implement C Program for performing following operations on Array: Insertion, Display.

## MCQ Questions

(Total number of Question=Marks\*3=06\*3=18)

1. Two main measures for the efficiency of an algorithm are
  - A. Processor and memory
  - B. Complexity and capacity
  - C. Time and space
  - D. Data and space
2. The time factor when determining the efficiency of algorithm is measured by
  - A. Counting microseconds
  - B. Counting the number of key operations
  - C. Counting the number of statements





- D. Counting the kilobytes of algorithm
3. The space factor when determining the efficiency of algorithm is measured by
- A. Counting the maximum memory needed by the algorithm**
  - B. Counting the minimum memory needed by the algorithm
  - C. Counting the average memory needed by the algorithm
  - D. Counting the maximum disk space needed by the algorithm
4. Which of the following case does not exist in complexity theory
- A. Best case
  - B. Worst case
  - C. Average case
  - D. Null case**
5. The Worst case occur in linear search algorithm when
- A. Item is somewhere in the middle of the array
  - B. Item is not in the array at all
  - C. Item is the last element in the array
  - D. Item is the last element in the array or is not there at all**
6. The Average case occur in linear search algorithm
- A. When Item is somewhere in the middle of the array**
  - B. When Item is not in the array at all
  - C. When Item is the last element in the array
  - D. When Item is the last element in the array or is not there at all
7. The complexity of the average case of an algorithm is
- A. Much more complicated to analyze than that of worst case
  - B. Much more simpler to analyze than that of worst case**
  - C. Sometimes more complicated and some other times simpler than that of worst case
  - D. None or above
8. The complexity of linear search algorithm is
- A.  $O(n)$**
  - B.  $O(\log n)$
  - C.  $O(n^2)$
  - D.  $O(n \log n)$
9. The complexity of Binary search algorithm is
- A.  $O(n)$
  - B.  $O(\log n)$**
  - C.  $O(n^2)$
  - D.  $O(n \log n)$
10. The complexity of Bubble sort algorithm is
- A.  $O(n)$
  - B.  $O(\log n)$
  - C.  $O(n^2)$**
  - D.  $O(n \log n)$



11. The complexity of merge sort algorithm is
- A.  $O(n)$
  - B.  $O(\log n)$
  - C.  $O(n^2)$
  - D.  $O(n \log n)$
12. The indirect change of the values of a variable in one module by another module is called
- A. Internal change
  - B. Inter-module change
  - C. **Side effect**
  - D. Side-module update
13. Which of the following data structure is not linear data structure?
- A. Arrays
  - B. Linked lists
  - C. Both of above
  - D. **None of above**
14. Which of the following data structure is linear data structure?
- A. Trees
  - B. Graphs
  - C. **Arrays**
  - D. None of above
15. The operation of processing each element in the list is known as
- A. Sorting
  - B. Merging
  - C. Inserting
  - D. **Traversal**
16. Finding the location of the element with a given value is:
- A. Traversal
  - B. **Search**
  - C. Sort
  - D. None of above
17. Arrays are best data structures
- A. **For relatively permanent collections of data**
  - B. For the size of the structure and the data in the structure are constantly changing
  - C. For both of above situation
  - D. For none of above situation
18. Linked lists are best suited
- A. for relatively permanent collections of data
  - B. **for the size of the structure and the data in the structure are constantly changing**
  - C. for both of above situation
  - D. for none of above situation
19. Each array declaration need not give, implicitly or explicitly, the information about
- A. The name of array
  - B. The data type of array
  - C. **The first data from the set to be stored**



- D. The index set of the array
20. The elements of an array are stored successively in memory cells because
- A. By this way computer can keep track only the address of the first element and the addresses of other elements can be calculated**
  - B. The architecture of computer memory does not allow arrays to store other than serially
  - C. Both of above
  - D. None of above



## 2. Searching and sorting

Position in Question Paper

Total Marks- 12

Q.2. a) 2-Marks.

Q.3. c) 4-Marks.

Q.4. a) 4-Marks.

Q.6. a) 6-Marks.

### Descriptive Questions

1. Define searching and give its type.
2. Define sorting and give its type.
3. Describe working of Linear Search with example.
4. Describe working of Binary Search with example.
5. Differentiate between Binary Search and Sequential Search.
6. Find the position of element 29 using binary search method in an array 'A' given below :  $A = \{11, 5, 21, 3, 29, 17, 2, 43\}$
7. Describe working of bubble sort with example.
8. Write an algorithm for Selection sort.  
Sort the following numbers in ascending order using Insertion sort.
9. Given Numbers: 348, 14, 614, 5381, 47.  
and Write the output after each iteration.
10. With explanation sort the given no. in ascending order using radix sort.  
Given Numbers: 348, 14, 614, 5381, 47.
11. Sort the following numbers in ascending order using quick sort.  
Given numbers 50, 2, 6, 22, 3, 39, 49, 25, 18, 5.
12. Sort the following numbers in ascending order using Bubble sort.  
Given Numbers: 348, 14, 614, 5381, 47  
and Write the output after each iteration.
13. Describe working of selection sort method. Also sort given input list in ascending order using selection sort input list : 55, 25, 5, 15, 35.



## MCQ Questions

(Total number of Question=Marks\*3=12\*3=36)

- 1) Which of the following is an external sorting?  
A. Insertion Sort  
B. Bubble Sort  
C. **Merge Sort**  
D. Tree Sort
- 2) Very slow way of sorting is.....  
A. **Insertion sort**  
B. Heap sort  
C. Bubble sort  
D. Quick sort
- 3) Which of the following is an internal sorting?  
A. Tape Sort  
B. 2-way Merge Sort  
C. Merge Sort  
D. **Tree Sort**
- 4) Sorting a file F usually refers to sorting F with respect to a particular key called.....  
A. Basic key  
B. **Primary key**  
C. Index key  
D. Starting key
- 5) The time complexity of quick sort is.....  
A. O(n)  
B. O(log n)  
C. O(n<sup>2</sup>)  
D. **O(n log n)**
- 6) Selection sort first finds them..... element in the list and put it in the first position.  
A. Middle element  
B. Largest element  
C. Last element  
D. **Smallest element**
- 7) Quick sort is also known as.....  
A. Merge sort  
B. Tree sort  
C. Shell sort  
D. **Partition and exchange sort**
- 8) The operation that combines the element is of A and B in a single sorted list C with n=r+s element is called....  
A. Inserting  
B. Mixing  
C. **Merging**  
D. Sharing
- 9) A tree sort is also known as.....sort.  
A. Quick  
B. Shell  
C. **Heap**  
D. Selection
- 10) sorting is good to use when alphabetizing large list of names.  
A. Merge  
C. **Radix**



- B. Heap  
11) The easiest sorting is.....  
A. Quick sort  
B. Shell sort  
12) Which of the following sorting algorithm is of divide and conquer type?  
A. Bubble sort  
B. Insertion sort  
13) Merging k sorted tables into a single sorted table is called.....  
A. **k way merging**  
B.  $k^{\text{th}}$  merge  
14) The function used to modify the way of sorting the keys of records is called.....  
A. Indexing function  
B. **Hash function**  
15) If the number of record to be sorted large and the key is short, then \_\_\_\_\_ sorting can be efficient.  
A. Merge  
B. Heap  
16) The total number of comparisons in a bubble sort is .....  
A.  $O(n \log n)$   
B.  $O(2n)$   
17) If the number of record to be sorted large and the key is long, then \_\_\_\_\_ sorting can be efficient.  
A. Merge  
B. Heap  
18) The time complexity of heap sort is .....  
A.  $O(n)$   
B.  $O(\log n)$   
19) The complexity of selection sort is .....  
A.  $O(n)$   
B.  **$O(n^2)$**   
20) Finding the location of a given item in a collection of items is called .....  
A. Discovering  
B. Finding  
21) The worst case occurs in linear search algorithm when.....  
D. Bubble  
C. Heap sort  
D. **Selection sort**  
C. Quick sort  
D. **Merge sort**  
C.  $k+1$  merge  
D.  $k-1$  merge  
C. Addressing function  
D. All of the above  
C. **Radix**  
D. Bubble  
C.  **$O(n^2)$**   
D.  $O(n)$   
C. **Quick**  
D. Bubble  
C.  **$O(n^2)$**   
D.  **$O(n \log n)$**   
C.  $O(n \log n)$   
D.  $O(\log n)$   
C. **Searching**  
D. Mining



- A. Item is somewhere in the middle of the array  
B. Item is not in the array at all  
C. Item is the last element in the array  
**D. Item is the last element in the array or item is not there at all**
- 22) If the number of records to be sorted is small, then \_\_\_\_\_ sorting can be efficient.  
A. Merge  
B. Heap  
C. Selection  
D. Bubble
- 23) The complexity of sorting algorithm measures the \_\_\_\_\_ as a function of the number  $n$  of items to be sorted.  
A. Average time  
B. Running time  
C. Average-case complexity  
D. Case-complexity
- 24) Which of the following is not a limitation of binary search algorithm?  
A. Must use a sorted array  
B. Sorted array is expensive when a lot of insertion and deletions are needed  
C. There must be a mechanism to access middle element directly  
**D. Binary search algorithm is not efficient when the data elements more than 1500.**
- 25) The Average case occurs in linear search algorithm \_\_\_\_\_ .  
A. when item is somewhere in the middle of the array  
B. when item is not the array at all  
C. when item is the last element in the array  
D. Item is the last element in the array or item is not there at all
- 26) Binary search algorithm cannot be applied to.....  
A. Sorted linked list  
B. Sorted binary trees  
C. Sorted linear array  
**D. Pointer array**
- 27) Complexity of linear search algorithm is.....  
A.  $O(n)$   
B.  $O(\log n)$   
C.  $O(n^2)$   
D.  $O(n \log n)$
- 28) Sorting algorithm can be characterized as.....  
A. Simple algorithm which require the order of  $n^2$  comparisons to sort  $n$  items.  
B. Sophisticated algorithms that require the  $O(n \log^2 n)$  comparisons to sort items.  
**C. Both of the above**  
D. None of the above
- 29) The complexity of bubble sort algorithm is \_\_\_\_\_ .  
A.  $O(n)$   
C.  $O(n^2)$



- B.  $O(\log n)$  D.  $O(n \log n)$
- 30) State True or False for internal sorting algorithms.
- Internal sorting are applied when the entire collection of data to be sorted is small enough that the sorting can take place within main memory.
  - The time required to read or write is considered to be significant in evaluating the performance of internal sorting.
- A. i-True, ii-True C. i-False, ii-True  
**B. i-True, ii-False** D. i-False, ii-False
- 31) The complexity of merge sort algorithm is.....
- A.  $O(n)$  C.  $O(n^2)$   
B.  $O(\log n)$  **D.  $O(n \log n)$**
- 32) \_\_\_\_\_ order is the best possible for array sorting algorithm which sorts  $n$  items.
- A.  $O(n \log n)$  **C.  $O(n + \log n)$**   
B.  $O(n^2)$  D.  $O(\log n)$
- 33) \_\_\_\_\_ is the method used by card sorter?
- A. Radix sort** C. Heap  
B. Insertion D. Quick
- 34) Sorting algorithm is frequently used when  $n$  is small where  $n$  is total number of elements.
- A. Heap C. Bubble  
**B. Insertion** D. Quick
- 35) Which of the following is not the required condition for binary search algorithm?
- A. The list must be sorted  
B. There should be the direct access to the middle element in any sub list  
**C. There must be mechanism to delete and/or insert elements in list.**  
D. Number values should only be present
- 36) Partition and exchange sort is \_\_\_\_\_.
- A. Quick sort** C. Heap sort  
B. Tree sort D. Bubble sort





### 3. Stacks and queues

Position in Question Paper

Total Marks- 20

Q.1. b) 2-Marks.

Q.1. d) 2-Marks.

Q.1. g) 2-Marks.

Q.2. c) 4-Marks.

Q.3. b) 4-Marks.

Q.4. d) 4-Marks.

Q.5. a) 6-Marks.

Q.6. b) 6-Marks.

#### Descriptive Questions

1. Enlist stack operations condition.
2. Differentiate between stack and queue. (any two points)
3. Show the memory representation of stack using array with the help of a diagram.
4. Give any two applications of stack.
5. Convert the following infix expression to its postfix form using stack  
 $A + B - C * D / E + F$ .
6. Convert infix expression into prefix expression :  $(A + B) * (C / G) + F$
7. Enlist queue operations condition.
8. Draw the diagram of queue to represent front and rear pointers of queue.
9. Sketch the diagram of circular queue.
10. List any 4 applications of queue.
11. Explain stack overflow and underflow conditions with example.  
Convert the following infix expression to its prefix form using stack
12.  $A + B - C * D / E + F$ .  
Show diagrammatically each step of conversion.
13. Convert following expression into postfix form. Give stepwise procedure.  
 $A + B \uparrow C * (D / E) - F / G$



- Show the effect of INSERT and DELETE operations on to the Linear queue of size 10. The Linear queue sequentially contains 10, 20, 30, 40, and 50 where 10 is at front
14. of the queue. Show diagrammatically the effect of-
- 1.INSERT(12)
  - 2.INSERT(34)
  3. DELETE
  4. INSERT (56)
15. Describe queue full and queue empty operation conditions on linear queue with suitable diagrams.
16. Write algorithm to insert an element into a linear queue
17. Write a program in 'C' to insert an element in a linear queue.
18. Draw and explain construction of circular queue.
- Show the effect of PUSH and POP operation on to the stack of size 10. The stack contains 40, 30, 52, 86, 39, 45, 50 with 50 being at top of the stack. Show diagrammatically the effect of :
19. (i) PUSH 59 (ii) PUSH 85 (iii) POP  
(iv) POP (v) PUSH 59 (vi) POP
- Sketch the final structure of stack after performing the above said operations.
20. Evaluate the following postfix expression :  
5, 6, 2, +, \*, 12, 4, /, - Show diagrammatically each step of evolution using stack.
21. Evaluate the following prefix expression :  
- \* + 4 3 2 5 show diagrammatically each step of evaluation using stack.
22. Write algorithm for performing push and pop operations on stack.
23. Define the term recursion. Write a program in C to display factorial of a entered number using recursion.

## MCO Questions

(Total number of Question=Marks\*3=20\*3=60)

- 1) \_\_\_\_\_form of access is used to add remove nodes from a stack.  
A. LIFO C. Both A and B  
B. FIFO D. None of these
- 2) New nodes are added to the \_\_\_\_\_ of the queue.  
A. Front C. Middle  
B. Back D. Both A and B
- 3) What happens when you push a new node onto a stack?  
A. The new node is placed at the front of the linked list



- B. The new node is placed at the back of the linked list  
C. The new node is placed at the middle of the linked list  
D. No Changes happens
- 4) Which of the following name does not relate to stacks?  
A. **FIFO lists** C. FILO list  
B. LIFO lists D. Push down lists
- 5) The term push and pop is related to  
A. Array C. **Stacks**  
B. Lists D. Trees
- 6) The elements are removal from a stack in \_\_\_\_ order.  
A. **Reverse** C. Alternative  
B. Hierarchical D. Sequential
- 7) \_\_\_\_ is the term used to insert an element into stack?  
A. **Push** C. Pop  
B. Pull D. Pump
- 8) \_\_\_\_ is the term used to delete an element from the stack?  
A. Push C. **Pop**  
B. Pull D. Pump
- 9) A pointer variable which contains the location at the top element of the stack is called.....  
A. **Top** C. Final  
B. Last D. End
- 10) Before deletion condition into stack \_\_\_\_\_ has to be checked.  
A. Overflow C. Maximum elements  
B. **Underflow** D. Existing elements
- 11) Before inserting into stack one must check the condition \_\_\_\_\_.  
A. **Overflow** C. Maximum elements  
B. Underflow D. Existing elements
- 12) When does Top value of stack change in insertion process?  
A. **Before insertion** C. At the time of insertion  
B. After insertion D. While checking overflow
- 13) Deletion in the linked stack takes place by deleting \_\_\_\_\_.  
A. **Node pointed by start process.** C. Beginning of the list  
B. End of the list D. Middle of the list
- 14) The condition \_\_\_\_\_ indicate the queue is empty.



- A. Front=Null  
B. Null=Front  
C. Front=Rear  
D. Rear=Null
- 15) The value of REAR is increased by 1 when \_\_\_\_\_.  
A. An element is deleted in a queue  
B. An element is traversed in a queue  
C. **An element is added in a queue**  
D. An element is merged in a queue
- 16) The term dequeue is the contraction of the name \_\_\_\_\_.  
A. **Double ended queue**  
B. Double side queue  
C. Double headed queue  
D. Double address queue
- 17) \_\_\_\_\_ is a collection of elements such that each element has been assigned a processing priority.  
A. **Priority queue**  
B. Procedure queue  
C. Main queue  
D. Interrupt queue
- 18) Link fields hold pointers to the \_\_\_\_\_ element in the linked representation of stack.  
A. **Neighboring**  
B. Last  
C. First  
D. Middle
- 19) Reversing a great deal of space for each stack in memory will \_\_\_\_\_.  
A. **Decrease the numbers of times overflow may occur**  
B. Increase the numbers of times overflow may occur  
C. Increase the number of times underflow may occur  
D. Increase the number of times underflow may occur.
- 20) Form of access is used to add and remove nodes from a queue.  
A. LIFO, Last In First Out  
B. **FIFO, First In First Out**  
C. Both a and b  
D. None of these
- 21) Entries in a stack are “ordered”. What is the meaning of this statement?  
A. A collection of stacks is sortable  
B. Stack entries may be compared with the ‘<’ operation  
C. The entries are stored in a linked list  
D. **There is a Sequential entry that is one by one**
- 22) Which of the following is not the application of stack?  
A. A parentheses balancing program  
B. Tracking of local variables at run time  
C. Compiler Syntax Analyzer  
D. **Data Transfer between two asynchronous process**
- 23) What is the value of the postfix expression 6 3 2 4 + - \*?



- A. 1  
B. 40  
C. 74  
D. -18
- 24) The postfix form of the expression  $(A + B * (C * D - E)) * F / G$  is?  
A.  $AB + CD * E - FG / **$   
B.  $AB + CD * E - F ** G /$   
C.  **$AB + CD * E - * F * G /$**   
D.  $AB + CDE * - * F * G /$
- 25) The data structure required to check whether an expression contains a balanced parenthesis is?  
A. **Stack**  
B. Queue  
C. Array  
D. Tree
- 26) What data structure would you mostly likely see in non-recursive implementation of a recursive algorithm?  
A. Linked List  
B. **Stack**  
C. Queue  
D. Tree
- 27) The postfix form of  $A * B + C / D$  is?  
A.  $* AB / CD +$   
B.  **$AB * CD / +$**   
C.  $A * BC + / D$   
D.  $ABCD + / *$
- 28) Which data structure is needed to convert infix notation to postfix notation?  
A. Branch  
B. Tree  
C. Queue  
D. **Stack**
- 29) The prefix form of  $A - B / (C * D ^ E)$  is?  
A.  $- / * ^ ACBDE$   
B.  $- ABCD * ^ DE$   
C.  **$- A / B * C ^ DE$**   
D.  $- A / BC * ^ DE$
- 30) What is the result of the following operation?  
**Top (Push (S, X))**  
A. **X**  
B.  $X + S$   
C. S  
D. XS
- 31) The prefix form of an infix expression  $(p + q) - (r * t)$  is?  
A.  $+ pq - * rt$   
B.  $- + pqr * t$   
C.  **$- + pq * rt$**   
D.  $- + * pqrt$
- 32) Which data structure is used for implementing recursion?  
A. Queue  
B. **Stack**  
C. Array  
D. List
- 33) The result of evaluating the postfix expression 5, 4, 6, +, \*, 4, 9, 3, /, +, \* is?



- A. 600  
**B. 350**  
C. 650  
D. 588

34) Consider the following operation performed on a stack of size 5.

Push(1);  
Pop();  
Push(2);  
Push(3);  
Pop();  
Push(4);  
Pop();  
Pop();  
Push(5);

After the completion of all operation, the number of elements present in stack is?

- A. 1**  
B. 2  
C. 3  
D. 4

35) Which of the following is not an inherent application of stack?

- A. Reversing a string  
B. Evaluation of postfix expression  
C. Implementation of recursion  
**D. Job scheduling**

36) The type of expression in which operator succeeds its operands is?

- A. Infix Expression  
B. Prefix Expression  
**C. Postfix Expression**  
D. Both Prefix and Postfix Expressions

37) If the elements "A", "B", "C" and "D" are placed in a stack and are deleted one at a time, what is the order of removal?

- A. ABCD  
**B. DCBA**  
C. DCAB  
D. ABDC

38) A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as \_\_\_\_\_.

- A. Queue**  
B. Stack  
C. Tree  
D. Linked list

39) The data structure required for Breadth First Traversal on a graph is?

- A. Stack  
B. Array  
**C. Queue**  
D. Tree

40) A queue follows \_\_\_\_\_.

- A. FIFO (First In First Out) principle**  
B. LIFO (Last In First Out) principle



- C. Ordered array  
D. Linear tree
- 41) Circular Queue is also known as \_\_\_\_\_.
- A. **Ring Buffer**
  - B. Square Buffer
  - C. Rectangle Buffer
  - D. Curve Buffer
- 42) If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time, in what order will they be removed?
- A. **ABCD**
  - B. DCBA
  - C. DCAB
  - D. ABDC
- 43) A data structure in which elements can be inserted or deleted at/from both ends but not in the middle is?
- A. Queue
  - B. Circular queue
  - C. **Deque**
  - D. Priority queue
- 44) A normal queue, if implemented using an array of size MAX\_SIZE, gets full when?
- A.  $\text{Rear} = \text{MAX\_SIZE} - 1$
  - B.  $\text{Front} = (\text{rear} + 1) \bmod \text{MAX\_SIZE}$
  - C.  $\text{Front} = \text{rear} + 1$
  - D.  $\text{Rear} = \text{front}$
- 45) Queues serve major role in \_\_\_\_\_.
- A. Simulation of recursion
  - B. Simulation of arbitrary linked list
  - C. **Simulation of limited resource allocation**
  - D. Simulation of heap sort
- 46) Which of the following is not the type of queue?
- A. Ordinary queue
  - B. **Single ended queue**
  - C. Circular queue
  - D. Priority queue
- 47) Which of the following real world scenarios would you associate with a stack data structure?
- A. **Piling up of chairs one above the other**
  - B. People standing in a line to be serviced at a counter
  - C. Offer services based on the priority of the customer
  - D. Tatkal Ticket Booking in IRCTC
- 48) What is the time complexity of pop() operation when the stack is implemented using an array?
- A. **O(1)**
  - B. O(n)
  - C. O(log n)
  - D. O(n log n)
- 49) How many stacks are required for evaluation of prefix expression?



- A. One  
**B. Two**
- C. Three  
D. Four
- 50) While evaluating a prefix expression, the string is read from?  
A. Left to right  
**B. Right to left**  
C. Center to right  
D. Center to left to right
- 51) What determines the order of evaluation of a prefix expression?  
**A. Precedence and associativity**  
B. Precedence only  
C. Associativity only  
D. Depends on the parser
- 52) Find the output of the following prefix expression  
 $*+2-2\ 1/-4\ 2+-5\ 3\ 1$   
**A. 2**  
B. 12  
C. 10  
D. 4
- 53) If implemented using an array of size MAX\_SIZE, gets full when \_\_\_\_\_.  
A.  $\text{Front} = (\text{rear} + 1) \bmod \text{MAX\_SIZE}$   
B.  $\text{Front} = \text{rear} + 1$   
**C.  $\text{Rear} = \text{MAX\_SIZE} - 1$**   
D.  $\text{Rear} = \text{front}$
- 54) Queues serve major role in  
A. Simulation of recursion  
B. Simulation of arbitrary linked list  
**C. Simulation of limited resource allocation**  
D. All of the mentioned
- 55) A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index.  
**A. 0**  
B. 7  
C. 9  
D. 10
- 56) In linked list implementation of queue, if only front pointer is maintained, which of the following operation take worst case linear time?  
A. Insertion  
B. Deletion  
C. To empty a queue  
D. Both Insertion and To empty a queue
- 57) Let the following circular queue can accommodate maximum six elements with the following data. What will happen after ADD O operation takes place?
- |                                |
|--------------------------------|
| front = 2 rear = 4             |
| queue = ___; L, M, N, ___, ___ |
- A. front = 2 rear = 5 queue = \_\_\_; L, M, N, O, \_\_\_**  
B. front = 3 rear = 5 queue = L, M, N, O, \_\_\_





**Maratha Vidya Prasarak Samaj's**  
**Rajarshi Shahu Maharaj Polytechnic, Nashik**

**Udoji Maratha Boarding Campus, Near Pumping Station, Gangapur Road, Nashik-13.**

**Affiliated to MSBTE Mumbai, Approved by AICTE New Delhi, DTE Mumbai & Govt. of Maharashtra, Mumbai.**

---

C. front = 3 rear = 4 queue = \_\_\_; L, M, N, O, \_\_\_

D. front = 2 rear = 4 queue = L, M, N, O, \_\_\_

58) Consider P,Q,R and S are the four elements in a queue. If we delete an element at a time then on which order they will get deleted?

**A. PQRS**

C. PSQR

B. SRQP

D. SRQP

59) A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index of\_\_\_.

**A. 0**

C. 9

B. 7

D. 10

60) In Queue, ENQUEUE means\_\_\_\_\_ whereas DEQUEUE refers\_\_\_\_\_.

**A. an insertion operation, a deletion operation.**

B. End of the queue, defining a queue.

C. Both A and B.

D. None of the above are true.



## 4. Linked lists

Position in Question Paper

Total Marks- 16

Q.2. b) 4-Marks.

Q.4. c) 4-Marks.

Q.4. e) 4-Marks.

Q.5. c) 6-Marks.

Q.6. c) 6-Marks.

### Descriptive Questions

1. Describe any two terms from the following :  
i) Node      ii) Null Pointer      iii) Empty List  
with respect to linear linked list with diagram.
2. Describe any two terms from the following :  
i) Information Field/ Data field      ii) Address      iii) Next Pointer  
with respect to circular linked list with diagram.
3. Write algorithm to delete an intermediate node from a Singly Linked List.
4. Write a program to traverse a linked list.
5. Write an algorithm to delete a node from the beginning of a circular linked list.
6. Describe circular linked list with suitable diagram. Also state advantage of circular linked list over linear linked list.
7. Create a Singly Linked List using data fields 10, 20, 30, 40, 50. Search a node 40 from the SLL and show procedure step-by-step with the help of diagram from start to end.
8. Describe procedure to delete an element from singly linked list using diagram.
9. Write an algorithm to insert an element at the beginning and at end of linked list.
10. Write an algorithm to count number of nodes in singly linked list.



## MCQ Questions

(Total number of Question=Marks\*3=16\*3=48)

- 1) Linked lists are best suited \_\_\_\_\_.  
A. For relatively permanent collections of data.  
**B. For the size of the structure and the data in the structure are constantly changing.**  
C. Data structure  
D. For none of above situation
- 2) The operation of processing each element in the list is known as \_\_\_\_\_.  
A. Sorting  
B. Merging  
C. Inserting  
**D. Traversal**
- 3) The situation when in a linked list START=NULL is \_\_\_\_\_.  
**A. Underflow**  
B. Overflow  
C. Houseful  
D. Saturated
- 4) Each node in singly linked list has \_\_\_\_\_ fields.  
**A. 2**  
B. 3  
C. 1  
D. 4
- 5) Which of the following is two way lists?  
A. Grounded header list  
B. Circular header list  
C. Linked list with header and trailer nodes  
**D. List traversed in two directions**
- 6) Which is the pointer associated with the availability list?  
A. FIRST  
**B. AVAIL**  
C. TOP  
D. REAR
- 7) Value of first linked list index is \_\_\_\_\_.  
**A. 0**  
B. 1  
C. -1  
D. 2
- 8) In linked lists there are no NULL links in  
A. Single linked list  
B. Linear doubly linked list  
**C. Circular linked list**  
D. Linked list
- 9) Each node in a linked list must contain at least \_\_\_\_\_.  
A. Three fields  
C. Four fields



- B. Two fields** D. Five fields
- 10) The dummy header in linked list contains \_\_\_\_\_.  
**A. First record of the actual data**  
B. Last record of the actual data  
C. Pointer to the last record of the actual data  
D. Middle record of the actual data
- 11) In a linked list the \_\_\_\_\_ field contains the address of next element in the list.  
**A. Link field** C. Start field  
B. Next element field D. Info field
- 12) LLINK is the pointer pointing to the \_\_\_\_\_.  
A. Successor node C. Head node  
**B. Predecessor node** D. Last node
- 13) \_\_\_\_\_ refers to a linear collection of data items.  
**A. List** C. Graph  
B. Tree D. Edge
- 14) A run list is \_\_\_\_\_.  
**A. Small batches of records from a file**  
B. Number of elements having same value  
C. Number of records  
D. Number of files in external storage
- 15) A \_\_\_\_\_ indicates the end of the list.  
A. Guard C. End pointer  
**B. Sentinel or NULL** D. Last pointer
- 16) A \_\_\_\_\_ is a linear list in which insertions and deletions are made to from either end of the structure.  
A. Circular queue C. Priority  
B. Random of queue **D. Dequeue**
- 17) Indexing the \_\_\_\_\_ element in the list is not possible in linked lists.  
**A. Middle** C. Last  
B. First D. Anywhere in between
- 18) A linear list in which the pointer points only to the successive node is \_\_\_\_\_.  
**A. Singly linked list** C. Doubly linked list  
B. Circular linked list D. None of the above
- 19) \_\_\_\_\_ may take place only when there is some minimum amount (or) no space left in free storage list.



- A. Memory management  
**B. Garbage collection**
- 20) A linear list in which the last node points to the first node is \_\_\_\_\_.  
A. Singly linked list  
**B. Circular linked list**  
C. Recycle bin  
D. Memory Allocation
- 21) To insert a new node in linked list free node will be available in \_\_\_\_\_.  
A. Available list  
**B. Avail list**  
C. Doubly linked list  
D. None of the above
- 22) A singly linked list is also called as \_\_\_\_\_.  
A. Linked list  
**B. One way chain**  
C. Free node list  
D. Memory space list
- 23) A \_\_\_\_\_ list is a header list where the node points back to the header node.  
**A. Circular header**  
B. Grounded header  
C. Two way chain  
D. Right link
- 24) A doubly linked list has \_\_\_\_\_ pointers with each node.  
A. 0  
B. 1  
**C. 2**  
D. 3
- 25) Header linked lists are frequently used for maintaining \_\_\_\_\_ in memory.  
**A. Polynomials**  
B. Binomial  
C. Trinomial  
D. Quadratic equation
- 26) The pointer that points to the first node in the list is \_\_\_\_\_.  
**A. FIRST**  
B. AVAIL  
C. TOP  
D. REAR
- 27) In a linked list, Deletion can be done as \_\_\_\_\_.  
A. Beginning  
B. End  
C. Middle  
**D. All of the above**
- 28) A doubly linked list is also called as \_\_\_\_\_.  
A. Linked list  
B. One way chain  
**C. Two way chain**  
D. Right link
- 29) The list that requires two pointer variables FIRST and LAST is called \_\_\_\_\_.  
A. Circular list  
B. Header list  
C. One way list  
**D. Two way list**
- 30) If the availability list is null, then the condition is said to be \_\_\_\_\_.  
A. Nil block  
C. Availability list overflow



- B. Availability list underflow** D. Memory loss
- 31) The list which has its own pointer is called \_\_\_\_\_.  
A. Pointer list C. Free pool  
B. Self-pointer D. Own pointer
- 32) Which of the following is two way lists?  
A. Grounded header list  
B. Circular header list  
C. Linked list with header and trailer nodes  
D. None of the above
- 33) A \_\_\_\_\_ is a header list where the last node contains the null pointer.  
A. Grounded header list C. Down header list  
B. Bottom header list D. Dropped header list
- 34) RLINK is the pointer pointing to the \_\_\_\_\_.  
A. Successor node C. Head node  
B. Predecessor node D. Last node
- 35) A \_\_\_\_\_ is a header list where the last node points back to the header node.  
A. Rounded header list C. Common header list  
B. Circular header list D. Forward header list
- 36) In a linked list, insertion can be done as \_\_\_\_\_.  
A. Beginning C. Middle  
B. End D. All of the above
- 37) In a two-way lists each node is divided into \_\_\_\_\_ parts.  
A. 1 C. 3  
B. 2 D. 4
- 38) The disadvantage in using a circular linked list is \_\_\_\_\_.  
A. It is possible to get into infinite loop  
B. Last node points to first node.  
C. Time consuming  
D. Requires more memory space.
- 39) Which of the following conditions checks available free space in avail list?  
A. Avail=NULL C. Avail=Max stack  
B. Null=Avail D. Avail=Top
- 40) A linear list in which each node has point to the predecessor and successors nodes is called \_\_\_\_\_.



- A. Singly linked list  
B. Circular linked list  
C. **Doubly linked list**  
D. Linear linked list
- 41) A linear collection of data elements where the linear node is given by means of pointer is called?  
A. **Linked list**  
B. Node list  
C. Primitive list  
D. Unordered list
- 42) Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in  $O(1)$  time?  
i) Insertion at the front of the linked list  
ii) Insertion at the end of the linked list  
iii) Deletion of the front node of the linked list  
iv) Deletion of the last node of the linked list  
A. I and II  
B. **I and III**  
C. I, II and III  
D. I, II and IV
- 43) In linked list each node contains a minimum of two fields. One field is data field to store the data second field is?  
A. Pointer to character  
B. Pointer to integer  
C. **Pointer to node**  
D. Node
- 44) What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?  
A.  $O(1)$   
B.  $O(n)$   
C.  **$\theta(n)$**   
D.  $\theta(1)$
- 45) What would be the asymptotic time complexity to insert an element at the front of the linked list (head is known)?  
A.  **$O(1)$**   
B.  $O(n)$   
C.  $O(n^2)$   
D.  $O(n^3)$
- 46) The concatenation of two lists can be performed in  $O(1)$  time. Which of the following variation of the linked list can be used?  
A. Singly linked list  
B. Doubly linked list  
C. **Circular doubly linked list**  
D. Array implementation of list
- 47) Consider the following definition in c programming language.

```
struct node
{
    int data;
```



```
struct node * next;  
}  
typedef struct node NODE;  
NODE *ptr;
```

Which of the following c code is used to create new node?

- A. **ptr = (NODE\*)malloc(sizeof(NODE));**  
B. ptr = (NODE\*)malloc(NODE);  
C. ptr = (NODE\*)malloc(sizeof(NODE\*));  
D. ptr = (NODE)malloc(sizeof(NODE));
- 48) Linked lists are not suitable for the implementation of \_\_\_\_\_  
A. Insertion sort  
B. Radix sort  
C. Polynomial manipulation  
D. **Binary search**
- 49) Linked list is considered as an example of \_\_\_\_\_ type of memory allocation.  
A. **Dynamic**  
B. Static  
C. Compile time  
D. Heap
- 50) Linked list data structure offers considerable saving in \_\_\_\_\_.  
A. Computational Time  
B. Space Utilization  
C. **Space Utilization and Computational Time**  
D. Speed Utilization
- 51) What does the following function do for a given Linked List with first node as head?  
**void fun1(struct node\* head)**  
{  
**if(head == NULL)**  
**return;**  
fun1(head->next);  
printf("%d ", head->data);  
}
- A. Prints all nodes of linked lists  
B. **Prints all nodes of linked list in reverse order**  
C. Prints alternate nodes of Linked List  
D. Prints alternate nodes in reverse order
- 52) Which of the following is not a disadvantage to the usage of array?





- A. Fixed size  
C. There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size  
C. Insertion based on position  
**D. Accessing elements at specified positions**
- 53) Which of these is not an application of a linked list?  
A. To implement file systems                      C. To implement non-binary trees  
B. For separate chaining in hash-tables        **D. Random Access of elements**
- 54) Which of the following is false about a doubly linked list?  
A. We can navigate in both the directions  
B. It requires more space than a singly linked list  
C. The insertion and deletion of a node take a bit longer  
**D. Implementing a doubly linked list is easier than singly linked list**
- 55) What is a memory efficient double linked list?  
**A. Each node has only one pointer to traverse the list back and forth**  
B. The list has breakpoints for faster traversal  
D. An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list  
D. A doubly linked list that uses bitwise AND operator for storing addresses
- 56) What is the worst case time complexity of inserting a node in a doubly linked list?  
A.  $O(n \log n)$     **C.  $O(n)$**   
B.  $O(\log n)$     D.  $O(1)$
- 57) What differentiates a circular linked list from a normal linked list?  
A. You cannot have the 'next' pointer point to null in a circular linked list  
B. It is faster to traverse the circular linked list  
**C. You may or may not have the 'next' pointer to null in a circular linked list**  
D. Head node is known in circular linked list
- 58) What is the time complexity of searching for an element in a circular linked list?  
**A.  $O(n)$**     C.  $O(1)$   
B.  $O(n \log n)$     D.  $O(n^2)$
- 59) Which of the following application makes use of a circular linked list?  
A. Undo operation in a text editor                      **C. Allocating CPU to resources**  
B. Recursive function calls                                D. Implement Hash Tables
- 60) Which of the following is false about a circular linked list?  
A. Every node has a successor



- B. Time complexity of inserting a new node at the head of the list is  $O(1)$**   
C. Time complexity for deleting the last node is  $O(n)$   
D. We can traverse the whole circular linked list by starting from any point
- 61) What is the best case time complexity of deleting a node in a Singly Linked list?  
A.  $O(n)$   
B.  $O(n^2)$   
C.  $O(n \log n)$   
**D.  $O(1)$**
- 62) What is the time complexity to insert a node based on key in a priority queue?  
A.  $O(n \log n)$   
B.  $O(\log n)$   
**C.  $O(n)$**   
D.  $O(n^2)$
- 63) What is not a disadvantage of priority scheduling in operating systems?  
A. A low priority process might have to wait indefinitely for the CPU  
B. If the system crashes, the low priority systems may be lost permanently  
**C. Interrupt handling**  
D. Indefinite blocking



## 5. Trees and graphs

Position in Question Paper

Total Marks- 16

Q.1. c) 2-Marks.

Q.1. e) 2-Marks.

Q.2. d) 4-Marks.

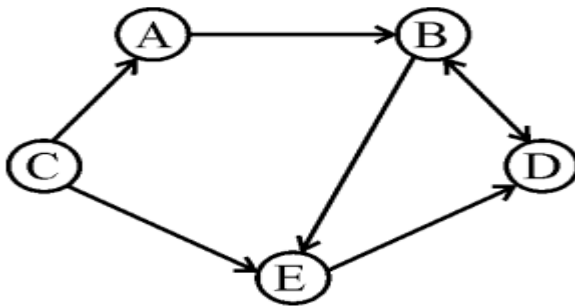
Q.3. d) 4-Marks.

Q.4. b) 4-Marks.

Q.5. b) 6-Marks.

### Descriptive Questions

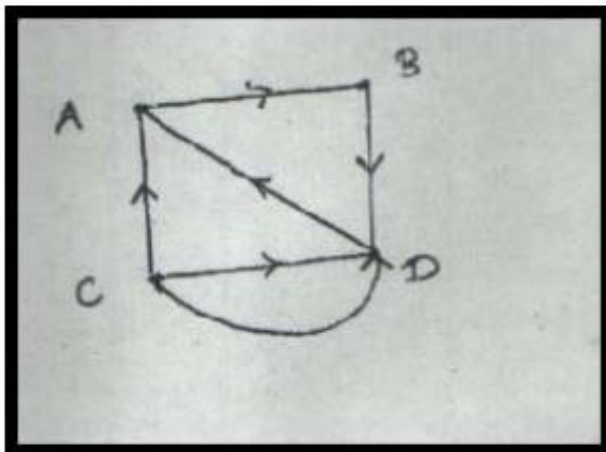
- Describe following terms w.r.to tree :
  - Leaf node
  - Level of node
- Describe any two types of trees from the following :
  - General Tree
  - Binary Tree
  - Binary Search Tree.
- State the following terms:
  - Ancestor
  - Height of Tree
- Define given two types of graph and give example.
  - Directed graph
  - Undirected graph
- Give any two applications of Graph.
- Differentiate between general tree and binary tree. (any four points)
- Construct the binary search tree using following elements:  
35,15,40,7,10,100,28,82,53,25,3.  
Show diagrammatically each step of construction of BST.
- Draw the tree structure of the following expressions :
  - $(2a + 5b)^3 * (x - 7y)^4$
  - $(a - 3b) * (2x - y)^3$
- Differentiate between tree and graph w.r.t. any 4 parameters.
- Explain Indegree and Outdegree of a graph with example.
- Give adjacency list and adjacency matrix for given graph :



For the following directed graph :

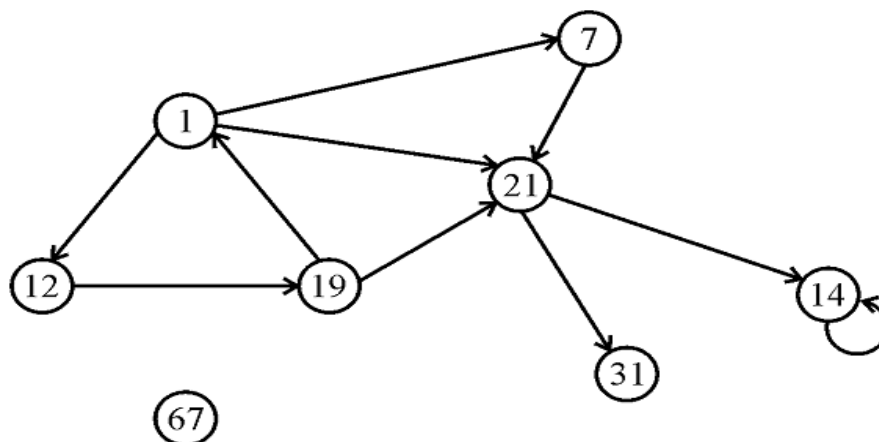
- i) Give adjacency matrix representation.
- ii) Give adjacency list representation.

12.



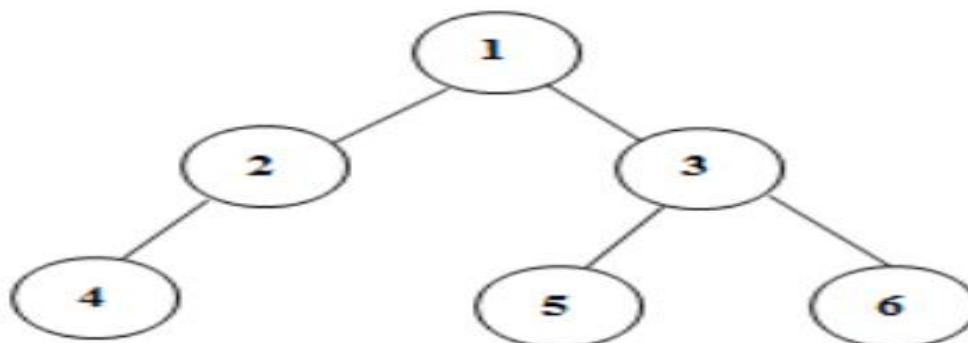
From the following graph, complete the answers :

13.



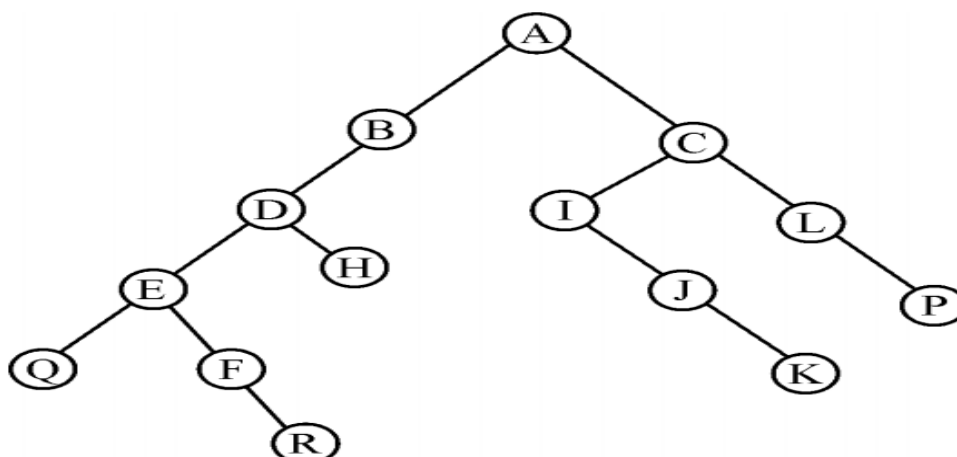
- (i) Indegree of node 21
- (ii) Adjacent node of 19
- (iii) Path of 31
- (iv) Successor of node 67

14. From the given tree complete six answers :



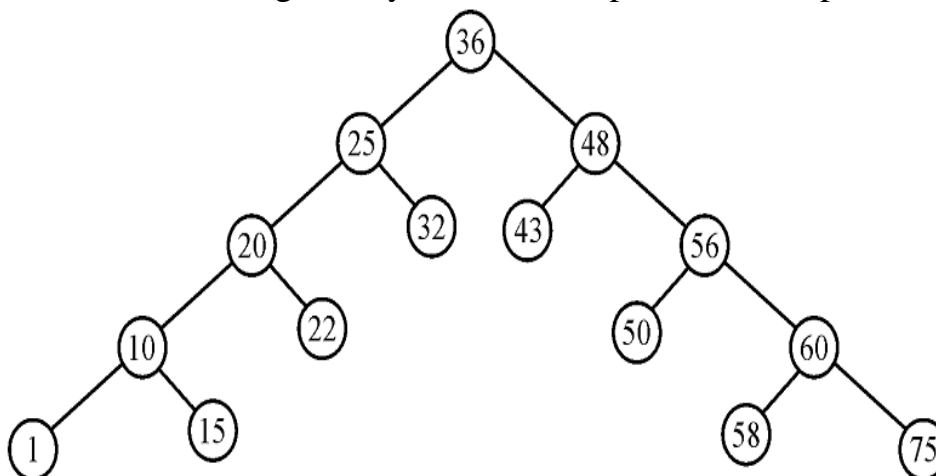
1. Degree of tree
  2. Degree of node 3
  3. Level of node 5
  4. Indegree of node 3
  5. Outdegree of node 3
  6. Height of tree
- For given binary tree write in-order, pre-order and post-order traversal.

15.



Traverse the following tree by the in-order, pre-order and post-order me

16.





## MCQ Questions

(Total number of Question=Marks\*3=16\*3=48)

- 1) Binary trees with threads are called as \_\_\_\_\_.  
A. Threaded trees  
B. Pointer trees  
C. Special trees  
D. Special pointer trees
- 2) In Binary trees nodes with no successor are called \_\_\_\_\_.  
A. End nodes  
B. Terminal nodes  
C. Final nodes  
D. Last nodes
- 3) Trees are said \_\_\_\_\_ if they are similar and have same contents at corresponding nodes.  
A. Duplicate  
B. Carbon copy  
C. Replica  
D. Copies
- 4) Every node N in a binary tree T except the root has a unique parent called the \_\_\_\_\_ of N.  
A. Antecedents  
B. Predecessor  
C. Forerunner  
D. Precursor
- 5) Sequential representation of binary tree uses \_\_\_\_\_.  
A. Array with pointers  
B. Single linear array  
C. Two dimensional arrays  
D. Three dimensional arrays
- 6) TREE [1] = NULL indicates tree is \_\_\_\_\_.  
A. Overflow  
B. Underflow  
C. Empty  
D. Full
- 7) Linked representation of binary tree needs \_\_\_\_\_ parallel arrays.  
A. 4  
B. 2  
C. 3  
D. 5
- 8) In a 2-tree, nodes with 0 children are called \_\_\_\_\_.  
A. Exterior node  
B. Outside node  
C. Outer node  
D. External node
- 9) In a extended-binary tree nodes with 2 children are called \_\_\_\_\_.  
A. Interior node  
B. Domestic node  
C. Internal node  
D. Inner node
- 10) The operation of processing each element in the list is known as \_\_\_\_\_.  
A. Sorting  
B. Merging  
C. Inserting  
D. Traversal

- 11) While converting binary tree into extended binary tree, all the original nodes in binary tree are \_\_\_\_.
- A. **Internal nodes on extended tree**                      C. Vanished on extended tree  
B. External nodes on extended tree                      D. Intermediate nodes on extended tree
- 12) In a binary tree, certain null entries are replaced by special pointers which point to nodes higher in the tree for efficiency. These special pointers are called \_\_\_\_.
- A. Leaf    C. Path  
B. Branch    D. **Thread**
- 13) The in order traversal of tree will yield a sorted listing of elements of tree in \_\_\_\_.
- A. Binary trees    C. Merging  
B. **Binary search trees**    D. AVL Trees
- 14) A binary tree whose every node has either zero or two children is called \_\_\_\_.
- A. Complete binary tree    C. **Extended binary tree**  
B. Binary Search tree    D. E2 tree
- 15) The post order traversal of a binary tree is DEBFC    A. Find out the pre order Traversal.
- A. ABFCDE    C. **ABDECF**  
B. ADBFEC    D. ABDCEF
- 16) In order traversing a tree resulted E A C K F H D B G; the preorder traversal would return.
- A. FAEKCDBHG    C. EAFKHDCBG  
B. **FAEKCDHGB**    D. FEAKDCHBG
- 17) \_\_\_\_ in linked representation of Binary trees LEFT[k] contains the \_\_\_\_ of at the node N, where k is the location.
- A. **Data**    C. Right child address  
B. Location and left child    D. Null value
- 18) Three standards ways of traversing a binary tree T with root R \_\_\_\_.
- A. Prefix, infix, postfix  
B. Pre-process, in-process, post-process  
C. Pre-traversal, in-traversal, post-traversal  
D. **Pre-order, in-order, post-order**
- 19) Which indicates pre-order traversal?
- A. Left sub-tree, Right sub-tree and root  
B. Right sub-tree, Left sub-tree and root  
C. **Root, Left sub-tree, Right sub-tree**

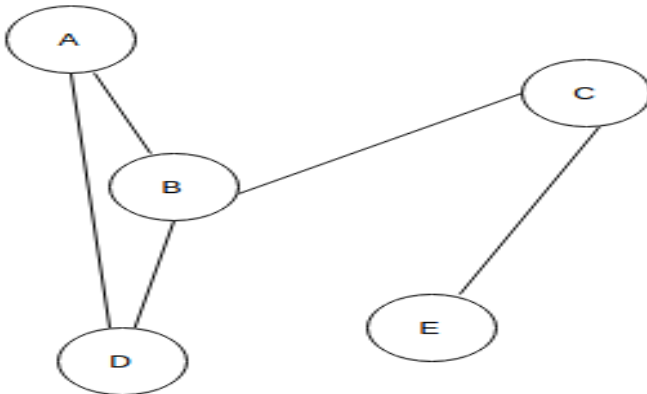


- D. Right sub-tree, root, Left sub-tree
- 20) A terminal node in a binary tree is called \_\_\_\_\_.  
A. Root  
**B. Leaf**  
C. Child  
D. Branch
- 21) Which indicates in-order traversal?  
A. Left sub-tree, Right sub-tree and root  
B. Right sub-tree, Left sub-tree and root  
C. Root, Left sub-tree, Right sub-tree  
**D. Right sub-tree, root, Left sub-tree**
- 22) The line drawn from a node N of tree T to a successor is called \_\_\_\_\_.  
A. Path  
**B. Edge**  
C. Arrow  
D. Route
- 23) In a binary tree a sequence of consecutive edges is called \_\_\_\_\_.  
A. Rotate  
B. Connecting lines  
C. Two-way  
**D. Path**
- 24) Which of the following indicates post-order traversal?  
**A. Left sub-tree, Right sub-tree and root**  
B. Right sub-tree, Left sub-tree and root  
C. Root, Left sub-tree, Right sub-tree  
D. Right sub-tree, root, Left sub-tree
- 25) The root R of the binary tree is assigned a level number of \_\_\_\_\_.  
A. 1  
**B. 0**  
C. -1  
D. Null
- 26) If node N is a terminal node in a binary tree then its \_\_\_\_\_.  
A. Right tree is empty  
B. Left tree is empty  
**C. Both left & right sub trees are empty**  
D. Root node is empty
- 27) In threaded binary tree \_\_\_\_\_ points to higher nodes in tree.  
A. Info  
B. Root  
**C. Threads**  
D. Child
- 28) A graph is said to be \_\_\_\_\_ if there is a path between any two of its nodes  
**A. Connected**  
B. Coupled  
C. Attached  
D. Allied





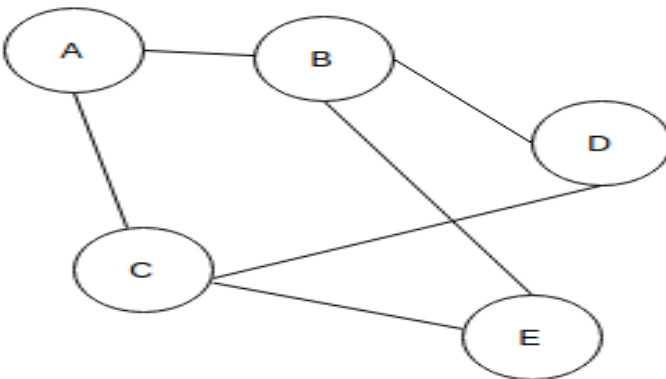
- 29) A graph is said to be \_\_\_\_\_ if every node  $u$  in  $G$  is adjacent to every other node  $v$  in  $G$ .
- A. Absolute  
B. Entire  
C. Inclusive  
**D. Complete**
- 30) A graph is said to be \_\_\_\_\_ if its edges are assigned data.
- A. Tagged  
B. Marked  
C. Labeled  
D. Sticked
- 31) Other name for directed graph is \_\_\_\_\_.
- A. Direct graph  
B. Die-graph  
C. Dia-graph  
**D. Digraph**
- 32) Graph  $G$  is \_\_\_\_\_ if for any pair  $u, v$  of nodes in  $G$  there is a path from  $u$  to  $v$  or path from  $v$  to  $u$ .
- A. Laterally connected  
B. Widely Connected  
C. **Un-literally connected**  
D. Literally connected
- 33) A connected graph  $T$  without any cycles is called \_\_\_\_\_.
- A. Free graph**  
B. No cycle graph  
C. Non cycle graph  
D. Circular graph
- 34) A connected graph  $T$  without any cycles is called a \_\_\_\_\_.
- A. A tree graph  
B. Free tree  
C. A tree d  
**D. All of the above**
- 35) In a graph if  $E = (u, v)$  means \_\_\_\_\_.
- A.  $u$  is adjacent to  $v$  but  $v$  is not adjacent to  $u$   
B.  $e$  begins at  $u$  and ends at  $v$   
C.  $u$  is processor and  $v$  is successor  
**D. Both b and c**
- 36) Which of the following statements for a simple graph is correct?
- A. Every path is a trail**  
B. Every trail is a path  
C. Every trail is a path as well as every path is a trail  
D. Path and trail have no relation
- 37) In the given graph identify the cut vertices.



- A. B and E
- B. C and D

- C. A and E
- D. C and B**

38) For the given graph (G), which of the following statements is true?



- A. G is a complete graph
- B. G is not a connected graph
- C. The vertex connectivity of the graph is 2**
- D. The edge connectivity of the graph is 1

39) What is the number of edges present in a complete graph having n vertices?

- A.  $(n*(n+1))/2$
- B.  $(n*(n-1))/2$**
- C. n
- D. Information given is insufficient

40) In a simple graph, the number of edges is equal to twice the sum of the degrees of the vertices.

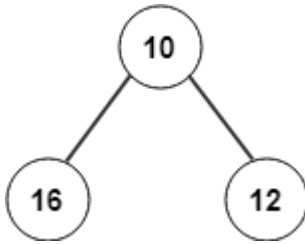
- A. True
- B. False**

41) Which of the following properties does a simple graph not hold?

- A. Must be connected**
- B. Must be un-weighted
- C. Must have no loops or multiple edges
- D. Must have no multiple edges



- 42) Which of the following is true?
- A. A graph may contain no edges and many vertices
  - B. A graph may contain many edges and no vertices**
  - C. A graph may contain no edges and no vertices
  - D. A graph may contain no vertices and many edges
- 43) For a given graph  $G$  having  $v$  vertices and  $e$  edges which is connected and has no cycles, which of the following statements is true?
- A.  $v=e$
  - C.  $v + 1 = e$
  - B.  $v = e+1$**
  - D.  $v = e-1$
- 44) A graph with all vertices having equal degree is known as a \_\_\_\_\_
- A. Multi Graph
  - C. Simple Graph
  - B. Regular Graph**
  - D. Complete Graph
- 45) Which of the following ways can be used to represent a graph?
- A. Adjacency List and Adjacency Matrix
  - B. Incidence Matrix
  - C. Adjacency List, Adjacency Matrix as well as Incidence Matrix**
  - D. No way to represent
- 46) How many children does a binary tree have?
- A. 2
  - C. **0 or 1 or 2**
  - B. any number of children
  - D. 0 or 1
- 47) What is/are the disadvantages of implementing tree using normal arrays?
- A. difficulty in knowing children nodes of a node
  - B. difficult in finding the parent of a node
  - C. have to know the maximum number of nodes possible before creation of trees**
  - D. difficult to implement
- 48) Advantages of linked list representation of binary trees over arrays?
- A. dynamic size
  - B. ease of insertion/deletion
  - C. ease in randomly accessing a node
  - D. both dynamic size and ease in insertion/deletion**
- 49) Which of the following traversing algorithm is not used to traverse in a tree?
- A. Post order
  - C. Post order
  - B. Pre order
  - D. Randomized**
- 50) The following given tree is an example for?



- A. Binary tree  
B. Binary search tree  
C. Fibonacci tree  
D. AVL tree

51) A binary tree is a rooted tree but not an ordered tree.

A. True

**B. False**

52) If binary trees are represented in arrays, what formula can be used to locate a left child, if the node has an index  $i$ ?

A.  $2i+1$

C.  $2i$

B.  $2i+2$

D.  $4i$

53) Which of the following properties are obeyed by all three tree – traversals?

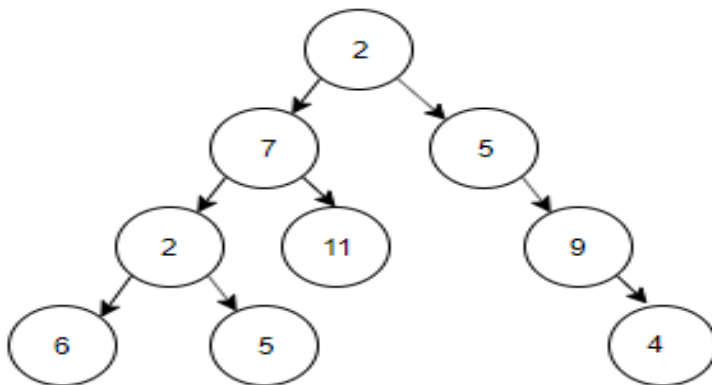
**A. Left sub-trees are visited before right sub-trees**

B. Right sub-trees are visited before left sub-trees

C. Root node is visited before left sub-tree

D. Root node is visited before right sub-tree

54) For the tree below, write the pre-order traversal.



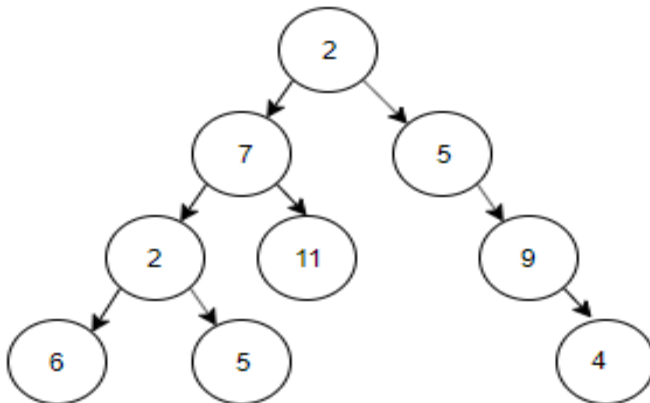
**A. 2, 7, 2, 6, 5, 11, 5, 9, 4**

C. 2, 5, 11, 6, 7, 4, 9, 5, 2

B. 2, 7, 5, 2, 6, 9, 5, 11, 4

D. 2, 7, 5, 6, 11, 2, 5, 4, 9

55) For the tree below, write the post-order traversal.



A. 2, 7, 2, 6, 5, 11, 5, 9, 4

C. 2, 5, 11, 6, 7, 4, 9, 5, 2

B. 2, 7, 5, 2, 6, 9, 5, 11, 4

D. 2, 7, 5, 6, 11, 2, 5, 4, 9

56) What is the time complexity of pre-order traversal in the iterative fashion?

A.  $O(1)$

C.  $O(\log n)$

**B.  $O(n)$**

D.  $O(n \log n)$

57) To obtain a prefix expression, which of the tree traversals is used?

A. Level-order traversal

C. Post-order traversal

**B. Pre-order traversal**

D. In-order traversal

58) In post-order traversal of binary tree right sub-tree is traversed before visiting root.

**A. True**

B. False

59) A full binary tree can be generated using \_\_\_\_\_

**A. post-order and pre-order traversal**

B. pre-order traversal

C. post-order traversal

D. in-order traversal

60) For a binary tree the first node visited in in-order and post-order traversal is same.

A. True

**B. False**