

**4300<sup>+</sup>**

Objective Solved  
Questions

**Volume-8**

Data Structures & Algorithms  
Computer Organization  
Operating System  
Communication  
Computer Network

# RSSB

**RAJASTHAN STAFF SELECTION BOARD**

**BASIC COMPUTER INSTRUCTOR**

*Subjectwise, Chapterwise Objective Solved Questions* ←

## Key Features

- ✦ Topic-wise Bifurcation of Questions
- ✦ Most of the Questions having Either Solution or Explanations.
- ✦ Also Useful for IA, NIC, BCI & other Computer Exams

National Board Helpline Number: + 91 809 444 1777

**EAPublications**  
ENGINEERS ACADEMY PUBLICATIONS

[W www.engineersacademy.org](http://www.engineersacademy.org)

[E info@engineersacademy.org](mailto:info@engineersacademy.org)



**Publisher and Distributor**

## **Engineers Academy Publications**

# 100-102, Ram Nagar, Bambala Puliya, Toll Tax,  
Tonk Road, Pratap Nagar, Jaipur (Rajasthan)-302033  
**E-Mail** : engineers.academy.india@gmail.com

### **All Rights Reserved :**

This book or part there of cannot be translated or reproduced in any form (except for review or criticism) without the written permission from the Publishers.

**ISBN : 978-93-93531-66-7**

**First Edition** : 2023  
**Second Edition** : 2026

Without prior written permission of publisher and author, no person/publisher/institute should use full part of the text/design/question/material of the book. If any body/publisher/institute is found in default legal action will be taken accordingly.

**Price : ₹ 200.00**

Although every effort has been made to avoid mistakes and omissions, there may be possibility some mistakes been left inadvertently. This book is released with the understanding that neither author nor publisher will be responsible in any manner for mistakes/premissions in the book. Dispute, if any, shall be subject to Jaipur (Rajasthan) Jurisdiction only.



## DIRECTOR'S *Message*

To reach heights one must start climbing and if the journey is difficult then perseverance is the key to success. As a teacher we have realized over past years that success in any competitive exam requires hard work and proper guidance. **Engineers Academy** with its unique teaching methodologies has always proved that we meet the expectations of thousands of students and parents to make their dreams come true. With changing patterns, we have adapted ourselves to deliver the best and ensure better results.

This book has been organized and executed with a lot of care, dedication and passion for lucidity. A conscious attempt has been made to simplify the concepts to facilitate better understanding of the subject.

Engineers Academy has many successful stories of students who secured All India Rank in ESE, GATE, PSUs and JEn. Now we invite you to become a part of Engineers Academy to explore and achieve ultimate goal of your life. We promise to provide you quality guidance with competitive environment which is far advanced and ahead than the reach of other institution.

We would feel satisfied if the book meets the needs of the students for whom it is meant.

Lastly, we are thankful to all the engineers, authors whose work has been the source of enlightenment, inspiration and guidance in presenting this book.

It is hoped that the book in its new form will enjoy its ever increasing popularity.

Regards

Dr. Pankaj Goyal



# Preface

✍ This book has been written to meet the growing requirements of candidates appearing for Basic Computer Instructor and other competitive Examinations. Though every candidate has ability to succeed but competitive environment, in-depth knowledge, quality guidance, time management and good source of study is required to achieve goals.

This book includes Multiple Choice Questions (MCQ's) which works as a mock exam practice for the reader. Questions of all the subject have been organized in systematic, concepts oriented and error less manner so that it become easy and interesting for even a beginner to understand. It is a very convenient book and must be solved by candidate aiming for competitive exams.

After solving this booklet students can feel encouraged and develop confidence to attempt each and every type of numerical as well as theoretical problems. Each problems explains solving approach so that at the end, so the reader is well equipped to be able to apply any type of problem solving requirement and distinctly choose one strategy or type from the other.

We hope this book will be proved an important tool to succeed in Basic Computer Instructor and other competitive Examinations.

Even though, enough readings were given for correcting the error and printing mistakes, due to human tendency there could be some minor typos in the book. If any such typos found, they will be highly appreciated and in corporated in the next edition. Also, please provide your valuable suggestions at :[engineers.academy.india@gmail.com](mailto:engineers.academy.india@gmail.com)

Wish you all the best. Have a nice reading.

**Team of  
Engineers Academy Publications**

# CONTENTS

## Volume 8

1. Data Structures & Algorithms.....	1 - 14
2. Computer Organization.....	15 - 28
3. Operating System.....	29 - 45
4. Communication.....	46 - 53
5. Computer Network.....	54 - 71

# DATA STRUCTURES & ALGORITHMS

## OBJECTIVE QUESTIONS

- What is a data structure/डेटा संरचना क्या है?
  - A programming language
  - A collection of algorithms
  - A way to store and organize data
  - A type of computer hardware
- How many types of Data Structure?/डेटा स्ट्रक्चर कितने प्रकार के होते हैं?
  - 2
  - 1
  - 3
  - 4
- Which of the following is Static data Structure/ निम्नलिखित में से कौन सा स्टेटिक डेटा स्ट्रक्चर है?
  - Tree
  - Graph
  - Array
  - Linked List
- Which of the following is Dynamic data Structure? / निम्नलिखित में से कौन सा डायनेमिक डेटा स्ट्रक्चर है?
  - Linked List
  - Stack
  - Queue
  - All of the Above
- What are the disadvantages of arrays? / एरे के नुकसान क्या हैं?
  - Array size is fixed and can't be changed after declaration
  - Insertion and deletion of elements are difficult
  - Memory may be wasted if the allocated size is larger than the required size
  - All of the above
- Which data structure is based on the Last In First Out (LIFO) principle/ कौन सी डेटा संरचना LIFO सिद्धांत पर आधारित है?
  - Tree
  - Linked List
  - Stack
  - Queue
- Which data structure is based on the First In First Out (FIFO) principle/ कौन-सी डेटा संरचना FIFO सिद्धांत पर आधारित है?
  - Tree
  - Linked List
  - Stack
  - Queue
- Which of the following is not the type of queue/निम्नलिखित में से कौन-सा क्यू का प्रकार नहीं है
  - Priority queue
  - Circular queue
  - Single ended queue
  - Ordinary queue
- What is the need for a circular queue/गोलाकार क्यू की क्या आवश्यकता है?
  - Easier computations
  - Implement LIFO principle in queues
  - Effective usage of memory
  - To delete elements based on priority
- A data structure in which elements can be inserted or deleted at/from both ends but not in the middle is? / एक डेटा संरचना जिसमें तत्वों को दोनों सिरों पर डाला या हटाया जा सकता है लेकिन बीच में नहीं?
  - Priority queue
  - Dequeue
  - Circular queue
  - Queue
- An algorithm is a \_\_\_\_\_ that provides a series of instructions that should be carried out in a particular order to get the desired outcome/एक एल्गोरिदम है जो निर्देशों की एक श्रृंखला प्रदान करता है जिसे वांछित परिणाम प्राप्त करने के लिए विशेष क्रम में पूरा किया जाना चाहिए?
  - Step by step process
  - Flow chart process
  - Pseudocode process
  - None of the above
- What are the advantages of Algorithm/एल्गोरिथ्म के क्या फायदे हैं?
  - Easy to understand
  - Algorithm follows a predefined path
  - It is independent of any programming language
  - All of the above
- Which of the following case does not exist in complexity theory/निम्नलिखित में से कौन सा मामला जटिलता सिद्धांत में मौजूद नहीं है?
  - Best case
  - Worst case
  - Average case
  - Null case
- The complexity of linear search algorithm is \_\_\_\_\_/ निम्न में से रेखिक खोज एल्गोरिदम की जटिलता ..... है
  - $O(n)$
  - $O(\log n)$
  - $O(n^2)$
  - $O(n \log n)$
- The complexity of Binary search algorithm is ..... / निम्न में से बाइनरी सर्च एल्गोरिदम की जटिलता..... है
  - $O(n)$
  - $O(\log n)$
  - $O(n^2)$
  - $O(n \log n)$
- The complexity of merge sort algorithm is \_\_\_\_\_ / निम्न में से मर्ज सॉर्ट एल्गोरिदम की जटिलता ..... है
  - $O(n)$
  - $O(\log n)$
  - $O(n^2)$
  - $O(n \log n)$
- The complexity of Bubble sort algorithm is ...../ निम्न में से बबल सॉर्ट एल्गोरिदम की जटिलता ..... है
  - $O(n)$
  - $O(\log n)$
  - $O(n^2)$
  - $O(n \log n)$

18. How many types of Asymptotic Notation/एसिम्प्टोटिक नोटेशन कितने प्रकार के होते हैं?  
 (a) 1 (b) 2  
 (c) 3 (d) 4
19. Which of the following is related to stack? / निम्नलिखित में से कौन स्टैक से संबंधित है  
 (a) Push (b) Pop  
 (c) LIFO (d) All of these
20. What type of data structure is Linked List/लिंकड लिस्ट किस प्रकार की डेटा संरचना है?  
 (a) Non-linear (b) Linear  
 (c) Hierarchical (d) None of these
21. Which of the following is Time Complexity Function/ निम्नलिखित में से कौन-सा समय जटिलता फंक्शन है?  
 (a) Constant (b) Linear  
 (c) Cubic (d) All the above
22. How many Types of Time Complexity Function/ समय जटिलता फंक्शन कितने प्रकार के होते हैं?  
 (a) 2 (b) 3  
 (c) 4 (d) 6
23. Which of the following is Linked List data Structure/ निम्नलिखित में से कौन लिंकड लिस्ट डेटा संरचना है?  
 (a) Singly (b) Doubly  
 (c) Circular (d) All the above
24. Which of the following is characteristic of algorithm/ निम्नलिखित में से कौन एल्गोरिथ्म की विशेषता है?  
 (a) Input (b) Output  
 (c) Finiteness (d) All of the Above
25. If there is no item in the list then what is it called/यदि सूची में कोई वस्तु नहीं है तो उसे क्या कहा जाता है?  
 (a) Null list (b) Empty list  
 (c) Zero List (d) None of these
26. What is the worst case complexity of quick sort / क्विक सॉर्ट की सबसे खराब स्थिति की जटिलता क्या है?  
 (a)  $O(n \log n)$  (b)  $O(n^2)$   
 (c)  $O(\log n)$  (d)  $O(n)$
27. In which of the following data structures can elements be deleted/ निम्नलिखित में से किस डेटा संरचना में तत्वों को हटाया जा सकता है?  
 (a) Stack (b) Queue  
 (c) Deque (d) Tree
28. Which of the following data structures stores homogeneous data items? / निम्नलिखित में से कौन सी डेटा संरचना समरूपी डेटा आइटम संग्रहीत करती है?  
 (a) Pointer (b) Array  
 (c) Record (d) None of these
29. If front = rear then queue is ..... / यदि front = rear है तो queue ..... है  
 (a) Full (b) Half full  
 (c) Empty (d) None of these
30. A graph is a collection of nodes called vertices and line segments called arcs or ..... that connect pairs of vertices./ एक ग्राफ नोड्स (वर्टिसेज) का संग्रह है और रेखाखंड जिन्हें आर्क या ..... कहा जाता है जो युग्मों को जोड़ते हैं।  
 (a) Vertices, edges (b) Edges, vertices  
 (c) Vertices, paths (d) Graph node, edges
31. A graph that has weights or costs associated with its edges is called ..... / वह ग्राफ जिसके किनारों के साथ भार या लागत जुड़ी हो उसे ..... कहते हैं  
 (a) Network (b) Weighted graph  
 (c) Both A and B (d) None A and B
32. State true or false: / सही या गलत बताएं  
 (i) The degree of root node is always zero.  
 (ii) Nodes that are not root and not leaf are called internal nodes.  
 (a) True, True (b) True, False  
 (c) False, True (d) False, False
33. A directed tree in which outdegree of each node is less than or equal to two is called ..... / एक निर्देशित ट्री जिसमें प्रत्येक नोड की आउटडिग्री दो से कम या बराबर होती है ..... कहलाता है  
 (a) Unary tree (b) Binary tree  
 (c) Trinary tree (d) Both B and C
34. State true or false: / सही या गलत बताएं  
 (i) An empty tree is also a binary tree.  
 (ii) In strictly binary tree, the out-degree of every node is either 0 or 2.  
 एक खाली (रिक्त) ट्री भी एक बाइनरी ट्री होता है।  
 सख्त बाइनरी ट्री में प्रत्येक नोड की आउट-डिग्री या तो 0 होती है या 2 होती है।  
 (a) True, False (b) False, True  
 (c) True, True (d) False, False
35. Operations on a data structure may be ..... / डेटा संरचना पर संचालन किस प्रकार हो सकता है .....?  
 (a) Creation (b) Destruction  
 (c) Selection (d) All of the above
36. The way in which data items are logically related defines ..... / जिस प्रकार डेटा आइटम तार्किक रूप से संबंधित हैं वह ..... को परिभाषित करता है  
 (a) Storage structure (b) Data structure  
 (c) Data relationship (d) Data operation
37. Which of the following operations are applicable on primitive data structures? / निम्नलिखित में से कौन से ऑपरेशन primitive डेटा संरचनाओं पर लागू होते हैं?  
 (a) Create (b) Destroy  
 (c) Update (d) All of the above
38. Which of the following data structure is non-linear type/ निम्नलिखित में से कौन सी डेटा संरचना गैर-रेखीय प्रकार की है?  
 (a) Strings (b) Lists  
 (c) Stacks (d) Tree

39. Each node in a linked list has two parts which is ...../ लिंकड लिस्ट में प्रत्येक नोड के कौनसे दो भाग होते हैं .....
- (a) Link field and information field – लिंक फील्ड और इंफॉर्मेशन फील्ड  
 (b) Link field and avail field – लिंक फील्ड और अवेल फील्ड  
 (c) Avail field and information field – अवेल फील्ड और इंफॉर्मेशन फील्ड  
 (d) Address field and link field – ,ड्रेस फील्ड और लिंक फील्ड
40. A linear list in which each node has pointers to its predecessor and successor nodes is called ...../एक रैखिक सूची जिसमें प्रत्येक नोड में पूर्ववर्ती और उत्तरवर्ती नोड्स के लिए संकेतक होते हैं ..... कहलाती है
- (a) Singly Linked List (b) Circular Linked List  
 (c) Doubly Linked List (d) Linear Linked List
41. Which of the following uses FIFO method?/निम्नलिखित में से कौन FIFO विधि का उपयोग करता है?
- (a) Queue (b) Stack  
 (c) Hash Table (d) Binary Search Tree
42. What data structure can be used to check if a syntax has balanced parenthesis/ Balanced parenthesis की जाँच के लिए किस डेटा संरचना का उपयोग किया जाता है?
- (a) Queue (b) Tree  
 (c) List (d) Stack
43. If the array is already sorted, which of these algorithms will exhibit the best performance/यदि एरे पहले से सॉर्टेड है, तो निम्न में से कौन सा एल्गोरिदम सर्वोत्तम प्रदर्शन करेगा?
- (a) Merge Sort (b) Insertion Sort  
 (c) Quick Sort (d) Heap Sort
44. Apriori algorithm analysis does not include/Apriori एल्गोरिदम विश्लेषण में क्या शामिल नहीं होता है?
- (a) Time Complexity (b) Space Complexity  
 (c) Program Complexity (d) None of the above
45. Which of the below mentioned sorting algorithms are not stable?/निम्नलिखित में से कौन सा सॉर्टिंग एल्गोरिदम स्थिर (Stable) नहीं है?
- (a) Selection Sort (b) Bubble Sort  
 (c) Merge Sort (d) Insertion Sort
46. Tower of Hanoi is a classic example of which programming paradigm/टॉवर ऑफ हनोई किस प्रोग्रामिंग प्रतिमान (Paradigm) का क्लासिक उदाहरण है?
- (a) Divide and Conquer (b) Greedy Algorithm  
 (c) Dynamic Programming (d) Randomized Algorithms
47. Pre-balancing of AVL tree costs/AVL ट्री में प्री-बैलेंसिंग की लागत क्या है?
- (a) 1 (b)  $(\log n)$   
 (c)  $n$  (d)  $n^2$
48. The worst case time complexity of AVL tree is better in comparison to binary search tree for/AVL ट्री की सबसे खराब स्थिति समय जटिलता, बाइनरी सर्च ट्री की तुलना में किसके लिए बेहतर है?
- (a) Search and insert operations  
 (b) Search and delete operations  
 (c) Insert and delete operations  
 (d) Search, insert and delete operations
49. Given an empty stack, after performing push(1), push(2), pop, push(3), push(4), pop, pop, what is the value of the top of the stack?/एक खाली स्टैक पर push(1), push(2), pop, push(3), push(4), pop, pop करने के बाद स्टैक के शीर्ष (top) का मान क्या होगा?
- (a) 4 (b) 3  
 (c) 2 (d) 1
50. Given two sorted lists of size 'm' and 'n' respectively. The number of comparisons needed in the worst case by the merge sort algorithm will be/आकार 'm' और 'n' की दो सॉर्टेड सूचियों के लिए मर्ज सॉर्ट एल्गोरिदम में सबसे खराब स्थिति में तुलना की संख्या क्या होगी?
- (a)  $m*n$  (b)  $\max(m, n)$   
 (c)  $\min(m, n)$  (d)  $m + n - 1$
51. Find the number of minimum comparisons required in the worst case to find both the minimum and the maximum value among n elements in an array./तत्वों में न्यूनतम और अधिकतम मान खोजने के लिए सबसे खराब स्थिति में न्यूनतम तुलना की संख्या क्या होगी?
- (a)  $2n + 2$  (b)  $n + 1$   
 (c)  $\text{floor}(3n/2) - 2$  (d)  $2n + \log n$
52. You have to sort a list L, consisting of a sorted list followed by a few 'random' elements. Which of the following sorting method would be most suitable for such a task?/एक सूची L जिसमें पहले से सॉर्टेड सूची है और उसके बाद कुछ 'रैंडम' तत्व हैं। ऐसे कार्य के लिए कौन सा सॉर्टिंग तरीका सबसे उपयुक्त है?
- (a) Bubble Sort (b) Selection Sort  
 (c) Quick Sort (d) Insertion Sort
53. Consider an array A[20, 10], assume 4 words per memory cell and the base address of array A is 100. What is the address of A[11,5]? Assume row major storage./एक एरे A[20,10], पर विचार करें। प्रति मेमोरी सेल 4 शब्द हैं और एरे A का बेस एड्रेस 100 है। Row major storage मानते हुए A[11,5] का पता क्या होगा?
- (a) 560 (b) 565  
 (c) 570 (d) 575
54. Level order traversal of a rooted tree can be done by starting from root and performing/रूटेड ट्री का लेवल ऑर्डर ट्रैवर्सल रूट से शुरू करके किसे करने से होता है?
- (a) Breadth First Search (b) Depth First Search  
 (c) Root Search (d) Deep Search
55. The average case occurs in the Linear Search Algorithm when:/लिनियर सर्च एल्गोरिदम में औसत स्थिति कब होती है?
- (a) The item to be searched is somewhere in the middle of the array खोजी जाने वाली वस्तु एरे के बीच में कहीं है  
 (b) The item to be searched is not in the array – खोजी जाने वाली वस्तु एरे में नहीं है  
 (c) The item to be searched is in the last of the array – खोजी जाने वाली वस्तु एरे के अंत में है  
 (d) The item to be searched is either in the last or not in the array खोजी जाने वाली वस्तु या तो एरे के अंत में है या एरे में नहीं है

56. Which of the following creates and utilize a symbol table?  
निम्नलिखित में से कौन सिंबल टेबल बनाता और उपयोग करता है?  
(a) Interpreter (b) Compiler  
(c) Loader (d) Assembler
57. Which of the following statement is NOT correct?/निम्नलिखित में से कौन सा कथन सही नहीं है?  
(a) Each activation record corresponds to a procedure/function./प्रत्येक एक्टिवेशन रिकॉर्ड एक प्रोसीजर.....फंक्शन से संबंधित होता है।  
(b) Symbol table uses array for implementation./सिंबल टेबल इम्प्लीमेंटेशन के लिए एरे का उपयोग करती है।  
(c) Symbol table is accessed by Syntax analyzer./सिंबल टेबल को सिंटैक्स एनालाइजर द्वारा एक्सेस किया जाता है।  
(d) Information on scope of a variable is stored in the symbol table./किसी वेरिएबल के स्कोप की जानकारी सिंबल टेबल में स्टोर की जाती है।
58. Dijkstra's algorithm is an example of \_\_\_\_\_/डिज्कस्ट्रा एल्गोरिद्म किसका उदाहरण है?  
(a) Divide and Conquer approach  
(b) Greedy approach  
(c) Dynamic programming approach  
(d) None of these
59. The data structure required to evaluate a postfix expression is \_\_\_\_\_/पोस्टफिक्स एक्सप्रेशन का मूल्यांकन करने के लिए कौन-सी डेटा संरचना आवश्यक है?  
(a) Queue (b) Stack  
(c) Array (d) Linked List
60. A mathematical model with a collection of operations defined on that model is called \_\_\_\_\_/एक गणितीय मॉडल जिसमें उस मॉडल पर परिभाषित संचालन का संग्रह हो, उसे ..... कहते हैं।  
(a) Data Structure (b) Abstract Datatype  
(c) Primitive Datatype (d) Algorithm
61. The minimum number of edges in a connected cyclic graph on  $n$  vertices is \_\_\_\_\_/शीर्षों वाले एक कनेक्टेड साइक्लिक ग्राफ में न्यूनतम किनारों की संख्या क्या होगी?  
(a)  $n + 1$  (b)  $n$   
(c)  $n - 1$  (d)  $2n$
62. Which of the following data structure is useful in implementing quick sort?/विषक सॉर्ट लागू करने में निम्नलिखित में से कौन सी डेटा संरचना उपयोगी है?  
(a) Stack (b) Set  
(c) List (d) Queue
63. The maximum degree of any vertex in a simple graph with  $n$  vertices is \_\_\_\_\_/न शीर्षों वाले एक सरल ग्राफ में किसी भी शीर्ष की अधिकतम डिग्री क्या होगी?  
(a)  $n$  (b)  $n - 1$   
(c)  $n + 1$  (d)  $2n + 1$
64. Which of the following expressions accesses the  $(i, j)$ th entry of a  $m \times n$  matrix stored in column major form?/ $m \times n$  मैट्रिक्स के  $(i, j)$  वें तत्व को column major form में एक्सेस करने का सही व्यंजक कौन सा है? रूप  
(a)  $n*(i + 1) + j$  (b)  $m*(j + 1) + i$   
(c)  $m*(n + j) + i$  (d)  $n*(m + i) + j$
65. A binary tree in which if all its levels except possibly the last have the maximum number of nodes and all the nodes at the last level appear as far left as possible, is called ...../एक बाइनरी ट्री जिसमें अंतिम स्तर को छोड़कर सभी स्तर अधिकतम नोड्स रखते हों और अंतिम स्तर के नोड्स बाएँ से भरे हों, उसे ..... कहते हैं।  
(a) Full Binary Tree (b) Binary Search Tree  
(c) Complete Binary Tree (d) Threaded Tree
66. A full binary tree with  $n$  leaves contains \_\_\_\_\_ nodes./ $n$  पत्तियों वाले एक पूर्ण बाइनरी ट्री में कुल कितने नोड्स होते हैं?  
(a)  $n$  (b)  $\log n$   
(c)  $2n - 1$  (d)  $2n + 1$
67. A 3-ary tree is a tree in which every internal node has exactly 3 children. The number of leaf nodes in such a tree with 6 internal nodes will be \_\_\_\_\_/3-एरी ट्री में प्रत्येक आंतरिक नोड के 3 child होते हैं। यदि 6 आंतरिक नोड हों, तो leaf नोड्स की संख्या कितनी होगी?  
(a) 10 (b) 9  
(c) 12 (d) 13
68. Traversing a binary tree first root and then left and right subtrees is called ..... traversal./बाइनरी ट्री में पहले रूट और फिर बाएँ तथा दाएँ subtrees का भ्रमण ..... ट्रैवर्सल कहलाता है।  
(a) Postorder (b) Preorder  
(c) Inorder (d) None of these
69. The number of nodes in a perfect binary tree of level 5 is ...../लेवल 5 के एक पूर्ण बाइनरी ट्री में नोड्स की संख्या कितनी होगी?  
(a) 15 (b) 20  
(c) 31 (d) 71
70. Number of possible binary trees with 3 nodes is .....? 3 नोड्स वाले संभव बाइनरी ट्री की संख्या कितनी है?  
(a) 12 (b) 9  
(c) 14 (d) 5
71. The time complexity of the linear search algorithm over an array of  $n$  elements is ...../न तत्वों की एरे पर लिनियर सर्च एल्गोरिद्म की समय जटिलता क्या है?  
(a)  $O(\log n)$  (b)  $O(n)$   
(c)  $O(n \log n)$  (d)  $O(n^2)$
72. Which of the following sorting algorithm does not have a worst case running time of  $O(n^2)$ ?/निम्नलिखित में से कौन सा सॉर्टिंग एल्गोरिद्म  $O(n^2)$  की worst case समय जटिलता नहीं रखता है?  
(a) Insertion Sort (b) Merge Sort  
(c) Quick Sort (d) Bubble Sort
73. Using the standard algorithm, what is the time required to determine that a number  $n$  is prime?/मानक एल्गोरिद्म का उपयोग करते हुए किसी संख्या  $n$  के अभाज्य होने का समय क्या है?  
(a) Linear time (b) Logarithmic time  
(c) Constant time (d) Quadratic time

74. To sort many large objects or structures, it would be most efficient to \_\_\_\_\_/कई बड़े ऑब्जेक्ट या संरचनाओं को सॉर्ट करने के लिए सबसे अधिक कुशल तरीका क्या होगा?
- Place reference to them in an array and sort the array
  - Place them in a linked list and sort the linked list
  - Place pointers to them in an array and sort the array
  - Place them in an array and sort the array
75. The average searching time of hashing, with linear probing, will be less if the load factor ...../लिनियर प्रोबिंग के साथ हैशिंग की औसत खोज समय कम होगा यदि लोड फैक्टर .....
- is far less than one
  - equals one
  - is far greater than one
  - None of these
76. Which of the following sorting procedure is slowest?/निम्नलिखित में से कौन सी सॉर्टिंग प्रक्रिया सबसे धीमी है?
- Quick sort
  - Merge sort
  - Shell sort
  - Bubble sort
77. Kruskal's algorithm for finding a minimum spanning tree of a weighted graph with  $n$  vertices and  $m$  edges has the time complexity of \_\_\_\_\_/n शीर्षों और  $m$  किनारों वाले भारित ग्राफ के न्यूनतम स्पैनिंग ट्री के लिए Kruskal एल्गोरिद्म की समय जटिलता क्या है?
- $O(n^2)$
  - $O(mn)$
  - $O(m + n)$
  - $O(m \log n)$
78. Which decision procedure has at least double exponential time complexity?/कौन-सी निर्णय समस्या कम से कम डबल एक्सपोनेंशियल समय जटिलता रखती है?
- Linear Programming
  - Travelling Salesman Problem
  - Presburger arithmetic
  - Hamiltonian circuit Problem
79. Which of the following shows the correct relationship among some of the more common computing times for algorithms?/निम्नलिखित में से कौन-सा एल्गोरिद्म की सामान्य समय जटिलताओं का सही क्रम दर्शाता है?
- $O(\log n) < O(n) < O(n \log n) < O(2n) < O(n^2)$
  - $O(n) < O(\log n) < O(n \log n) < O(2n) < O(n^2)$
  - $O(n) < O(\log n) < O(n \log n) < O(n^2) < O(2n)$
  - $O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2n)$
80.  $f(n)$  is of the order of  $g(n)$  if there exist positive integers "a" and "b" such that: यदि धनात्मक पूर्णांक "a" और "b" मौजूद हों तो  $f(n)$ ,  $g(n)$  के क्रम का होगा जब
- $f(n) = b$
  - $f(n) d^a \cdot g(n)$  for all  $n e^b$
  - $g(n) d^a \cdot f(n)$  for all  $n e^b$
  - None of these
81. " $O(k)$ , where  $O(n)$  stands for order  $n$  is \_\_\_\_\_"/ " $O(k)$ , जहाँ  $O(n)$  का अर्थ order  $n$  है, होगा \_\_\_\_\_
- $O(n)$
  - $O(n^2)$
  - $O(n^3)$
  - $O(3n^3)$
82. Which of the following statements is true?/निम्नलिखित में से कौन सा कथन सत्य है?
- As the number of entries in a hash table increases, the number of collisions increases.
  - Recursive programs are efficient.
  - The worst case complexity for Quicksort is  $O(n^2)$ .
  - Binary search using a linear linked list is efficient.
- 1 and 2
  - 2 and 3
  - 1 and 4
  - 1 and 3
83. The concept of order (Big O) is important because ...../ऑर्डर (Big O) की अवधारणा महत्वपूर्ण है क्योंकि .....
- It can be used to decide the best algorithm that solves a given problem
  - It determines the maximum size of a problem that can be solved in a given amount of time
  - It is the lower bound of the growth rate of algorithm
  - Both (a) and (b)
84. Which of the following is false?/निम्नलिखित में से कौन सा कथन गलत है?
- $100n \log n = O(n \log n)$
  - $+(\log n) = O(\log \log n)$
  - If  $0 < x < y$  then  $n^x = O(n^y)$
  - $2n^x + O(nk)$
85. The concatenation of two lists is to be performed in  $O(1)$  time. Which of the following implementations of a list should be used?/दो सूचियों का संयोजन  $O(1)$  समय में करना है। निम्न में से कौन सा कार्यान्वयन उपयोग करना चाहिए?
- Singly linked list
  - Doubly linked list
  - Circular doubly linked list
  - Array implementation of list
86. Consider the following three claims:/निम्नलिखित तीन दावों पर विचार करें
- $(n + k)^m = O(m)$ , where  $k$  and  $m$  are constants
  - $2n^2 = O(2n)$
  - $2n! = O(2n)$
- Which of these claims are correct?/इनमें से कौन से कथन सही हैं?
- 1 and 2
  - 1 and 3
  - 2 and 3
  - 1, 2 and 3
87. The tightest lower bound on the number of comparisons, in the worst case, for comparison based sorting is of the order of \_\_\_\_\_/तुलना आधारित सॉर्टिंग के लिए worst case में तुलना की संख्या का सबसे सख्त निचला बाउंड ..... है।
- $n$
  - $n \log n$
  - $n \log, n$
  - $n \log, n$

88. The time complexity of the following C function is (assume  $n > 0$ ) / निम्न C फंक्शन की समय जटिलता क्या है? (मान लें  $n > 0$ )
- ```
int recursive (int n)
{
if(n == 1)
return(1);
else
return(recursive(n-1) + recursive(n-1));
}
```
- (a)  $O(n)$  (b)  $O(n \log n)$   
(c)  $O(n^2)$  (d)  $O(2n)$
89. The time complexity of computing the transitive closure of a binary relation on a set of  $n$  elements is known to be \_\_\_\_\_ /  $n$  तत्वों के सेट पर बाइनरी रिलेशन की ट्रांजिटिव क्लोजर निकालने की समय जटिलता ..... है।
- (a)  $O(n)$  (b)  $O(n \log n)$   
(c)  $O(n^3)$  (d)  $O(n^2)$
90. The minimum number of comparisons required to determine if an integer appears more than  $n/2$  times in a sorted array of  $n$  integers is ..... /  $n$  पूर्णाकों की सॉर्टेड एरे में यह निर्धारित करने के लिए कि कोई संख्या  $n/2$  से अधिक बार आती है, न्यूनतम तुलना की संख्या ..... है।
- (a)  $O(n)$  (b)  $O(\log n)$   
(c)  $O(\log^2 n)$  (d)  $\Theta(1)$
91. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity? / निम्नलिखित में से कौन सा सॉर्टिंग एल्गोरिद्म एक रैंडम लिंकड लिस्ट को न्यूनतम समय जटिलता में सॉर्ट करने के लिए उपयोग किया जा सकता है?
- (a) Insertion Sort (b) Quick Sort  
(c) Heap Sort (d) Merge Sort
92. What does the following function do for a given Linked List with first node as head? / निम्नलिखित फंक्शन, जिसमें पहला नोड head है, लिंकड लिस्ट के लिए क्या करता है?
- ```
void fun1(struct node* head)
{
if(head == NULL)
return;
fun1(head->next);
printf("%d", head->data);
}
```
- (a) Prints all nodes of linked list  
(b) Prints all nodes of linked list in reverse order  
(c) Prints alternate nodes of Linked List  
(d) Prints alternate nodes in reverse order
93. Which of the following points is/are true about Linked List data structure when it is compared with array? / एरे की तुलना में लिंकड लिस्ट डेटा संरचना के बारे में निम्नलिखित में से कौन सा कथन सही है?
- (a) It is easy to insert and delete elements in Linked List  
(b) Random access is not allowed in a typical implementation of Linked Lists  
(c) The size of array has to be pre-decided, linked lists can change their size any time  
(d) All of the above
94. In the worst case, the number of comparisons needed to search a singly linked list of length  $n$  for a given element is \_\_\_\_\_? सबसे खराब स्थिति में, लंबाई  $n$  की सिंगली लिंकड लिस्ट में किसी तत्व को खोजने के लिए तुलना की संख्या ..... होगी।
- (a)  $\log_2 n$  (b)  $n/2$   
(c)  $\log_2 n - 1$  (d)  $n$
95. In a singly linked list, which operation takes the maximum time complexity when the pointer to the head node is given? / एक सिंगली लिंकड लिस्ट में, जब हेड नोड का पॉइंटर दिया गया हो, तो निम्न में से कौन-सा ऑपरेशन अधिकतम समय जटिलता लेता है?
- (a) Insertion at beginning  
(b) Deletion at beginning  
(c) Traversing the entire list  
(d) Accessing the first node
96. What are the time complexities of finding 8th element from beginning and 8th element from end in a singly linked list? Let  $n$  be the number of nodes in linked list, you may assume that  $n > 8$ . / सिंगली लिंकड लिस्ट में शुरुआत से 8वाँ और अंत से 8वाँ तत्व खोजने की समय जटिलता क्या होगी? (मान लें  $n > 8$ )
- (a)  $O(1)$  and  $O(n)$  (b)  $O(1)$  and  $O(1)$   
(c)  $O(n)$  and  $O(1)$  (d)  $O(n)$  and  $O(n)$
97. Given pointer to a node  $X$  in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete node  $X$  from given linked list? / सिंगली लिंकड लिस्ट में नोड  $X$  का पॉइंटर दिया गया है, हेड का पॉइंटर नहीं दिया गया। क्या हम  $X$  को डिलीट कर सकते हैं?
- (a) Possible if  $X$  is not last node. Use two steps:  
(i) Copy the data of next of  $X$  to  $X$ .  
(ii) Delete next node of  $X$ .  
(b) Possible if size of linked list is even.  
(c) Possible if size of linked list is odd.  
(d) Possible if  $X$  is not first node.
98. You are given pointers to first and last nodes of a singly linked list, which of the following operations are independent of the length of the linked list? / यदि सिंगली लिंकड लिस्ट के पहले और अंतिम नोड के पॉइंटर दिए गए हों, तो निम्न में से कौन सा ऑपरेशन लिस्ट की लंबाई पर निर्भर नहीं करता?
- (a) Delete the first element  
(b) Insert a new element as first element  
(c) Delete the last element  
(d) Add a new element at the end of the list
99. The concatenation of two lists is to be performed in  $O(1)$  time. Which of the following implementations of a list should be used? / दो सूचियों का संयोजन  $O(1)$  समय में करना है। निम्न में से कौन सा कार्यान्वयन उपयोग करना चाहिए?

- (a) Singly linked list  
 (b) Doubly linked list  
 (c) Circular doubly linked list  
 (d) Array implementation of lists
100. In a doubly linked list, the number of pointers affected for an insertion operation will be \_\_\_\_\_/डबल लिंकड लिस्ट में एक इंसर्शन ऑपरेशन के दौरान प्रभावित पॉइंटर्स की संख्या कितनी होगी?  
 (a) 4 (b) 0  
 (c) 1 (d) None of these
101. Which one of the following is an application of Stack Data Structure?/निम्नलिखित में से कौन सा स्टैक डेटा संरचना का अनुप्रयोग है?  
 (a) Managing function calls  
 (b) Tower of Hanoi problem  
 (c) Arithmetic expression evaluation  
 (d) All of the above
102. Which of the following is true about linked list implementation of stack/स्टैक के लिंकड लिस्ट इम्प्लीमेंटेशन के बारे में निम्नलिखित में से कौन सा कथन सही है?  
 (a) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.  
 (b) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from beginning.  
 (c) Both of the above  
 (d) None of the above
103. The result evaluating the postfix expression  $10\ 5 + 60\ 6 / * 8$  - is \_\_\_\_\_/पोस्टफिक्स एक्सप्रेशन  $10\ 5 + 60\ 6 / * 8$  - का परिणाम क्या होगा?  
 (a) 284 (b) 213  
 (c) 142 (d) 71
104. Suppose a stack is to be implemented with a linked list instead of an array. What would be the effect on the time complexity of the push and pop operations of the stack (assuming stack is implemented efficiently)?/यदि स्टैक को एरे के बजाय लिंकड लिस्ट से लागू किया जाए, तो push और pop ऑपरेशन की समय जटिलता पर क्या प्रभाव पड़ेगा? (मान लें स्टैक कुशलता से लागू किया गया है)  
 (a)  $O(1)$  for insertion and  $O(n)$  for deletion  
 (b)  $O(1)$  for insertion and  $O(1)$  for deletion  
 (c)  $O(n)$  for insertion and  $O(1)$  for deletion  
 (d)  $O(n)$  for insertion and  $O(n)$  for deletion
105. If the sequence of operations – push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2), pop are performed on a stack, the sequence of popped out values is \_\_\_\_\_/यदि स्टैक पर निम्न ऑपरेशन किए जाएँ – push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2), pop, तो pop किए गए मानों का क्रम क्या होगा?  
 (a) 2, 2, 1, 1, 2 (b) 2, 2, 1, 2, 2  
 (c) 2, 1, 2, 2, 1 (d) 2, 1, 2, 2, 2
106. Consider the following operations performed on a stack of size 5:  
 Push(a); Pop(); Push(b); Push(c); Pop(); Push(d); Pop(); Pop(); Push(e)  
 Which of the following statements is correct?  
 आकार 5 के स्टैक पर निम्न ऑपरेशन किए गए हैं  
 Push(a); Pop(); Push(b); Push(c); Pop(); Push(d); Pop(); Pop(); Push(e)  
 निम्न में से कौन सा कथन सही है?  
 (a) Underflow occurs  
 (b) Stack operations are performed smoothly  
 (c) Overflow occurs  
 (d) None of the above
107. Which of the following is true about linked list implementation of queue?/Queue के लिंकड लिस्ट इम्प्लीमेंटेशन के बारे में निम्नलिखित में से कौन सा कथन सही है?  
 (a) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.  
 (b) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from beginning.  
 (c) Both of the above  
 (d) None of the above
108. Postorder traversal of a given binary search tree T produces the following sequence of keys:  
 10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29  
 Which one of the following sequence of keys can be the result of an in-order traversal of the tree T?  
 एक बाइनरी सर्च ट्री T का पोस्टऑर्डर ट्रैवर्सल निम्न क्रम देता है  
 10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29  
 निम्न में से कौन सा क्रम T के इन-ऑर्डर ट्रैवर्सल का परिणाम हो सकता है?  
 (a) 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95  
 (b) 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29  
 (c) 9, 10, 15, 22, 23, 27, 40, 60, 50, 25, 29, 95  
 (d) 95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29
109. Consider a node X in a binary tree. Given that X has two children, let Y be inorder successor of X. Which of the following is true about Y?/यदि X के दो child हैं और Y उसका inorder successor है, तो Y के बारे में कौन सा कथन सही है?  
 (a) Y has no right child (b) Y has no left child  
 (c) Y has both children (d) None of the above
110. The height of a tree is the length of the longest root-to-leaf path. The maximum and minimum number of nodes in a binary tree of height 5 are \_\_\_\_\_ respectively./ऊँचाई 5 वाले बाइनरी ट्री में अधिकतम और न्यूनतम नोड्स की संख्या क्या होगी?  
 (a) 63 and 6 (b) 64 and 5  
 (c) 32 and 6 (d) 31 and 5

111. What is the worst case time complexity for search, insert and delete operations in a general Binary Search Tree?/सामान्य BST में search, insert और delete की worst case समय जटिलता क्या है?
- (a)  $O(n)$  for all  
 (b)  $O(\log n)$  for all  
 (c)  $O(\log n)$  for search and insert,  $O(n)$  for delete  
 (d)  $O(\log n)$  for search, and  $O(n)$  for insert and delete
112. Which of the following traversal outputs the data in sorted order in a BST?/BST में कौन सा traversal डेटा को sorted order में देता है?
- (a) Preorder (b) Inorder  
 (c) Postorder (d) Level order
113. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted into an initially empty BST. What is the in-order traversal sequence?/यदि दिए गए क्रम में संख्याएँ BST में डाली जाएँ, तो उसका inorder traversal क्या होगा?
- (a) 7 5 1 0 3 2 4 6 8 9 (b) 0 2 4 3 1 6 5 9 8 7  
 (c) 0 1 2 3 4 5 6 7 8 9 (d) 9 8 6 4 2 3 0 1 5 7
114. The following numbers are inserted into an empty BST in the given order:  
 10, 1, 3, 5, 15, 12, 16.  
 What is the height of the BST?  
 दिए गए क्रम में BST में नोड्स डालने पर ट्री की ऊँचाई क्या होगी?
- (a) 2 (b) 3  
 (c) 4 (d) 6
115. While inserting 71, 65, 84, 69, 67, 83 in an empty BST, the element in the lowest level is \_\_\_\_\_./दिए गए क्रम में BST में डालने पर सबसे निचले स्तर पर कौन सा तत्व होगा?
- (a) 65 (b) 67  
 (c) 69 (d) 83
116. A BST is generated by inserting:  
 50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24  
 The number of nodes in the left subtree and right subtree of the root respectively is \_\_\_\_\_.  
 दिए गए क्रम से बने BST में root के left subtree और right subtree में क्रमशः कितने नोड होंगे?
- (a) 6 and 5 (b) 5 and 6  
 (c) 7 and 4 (d) 4 and 7
117. The following numbers are inserted into an empty binary search tree in the given order:  
 10, 1, 3, 5, 15, 12, 16.  
 What is the height of the binary search tree?  
 दिए गए क्रम में BST में नोड्स डालने पर ट्री की ऊँचाई क्या होगी?
- (a) 3 (b) 4  
 (c) 5 (d) 6
118. Which of the following is an advantage of adjacency list representation over adjacency matrix representation of a graph?/ग्राफ के adjacency matrix की तुलना में adjacency list का कौन सा लाभ है?
- (a) In adjacency list representation, space is saved for sparse graphs  
 (b) DFS and BFS can be done in  $O(V + E)$  time in adjacency list representation  
 (c) Adding a vertex in adjacency list representation is easier  
 (d) All of the above
119. Which of the following statements is/are TRUE for an undirected graph?/  
 P: Number of odd degree vertices is even  
 Q: Sum of degrees of all vertices is even  
 अनिर्देशित ग्राफ के लिए कौन सा कथन सही है?
- (a) P Only (b) Q Only  
 (c) Both P and Q (d) Neither P nor Q
120. Which of the following array element will return the top-of-the-stack-element for a stack of size N elements(capacity of stack > N)?/निम्नलिखित में से कौन-सा array (सरणी) तत्व, N तत्वों वाले stack (जहाँ stack capacity, N से अधिक है) के लिए top-of-the-stack element को return करेगा?
- (a)  $S[N-1]$  (b)  $S[N]$   
 (c)  $S[N-2]$  (d)  $S[N+1]$
121. How many undirected graphs (not necessarily connected) can be constructed out of a given set  $V = \{V_1, V_2, V_3, \dots, V_n\}$  of n vertices?/n शीर्षों के समुच्चय  $V = \{V_1, V_2, V_3, \dots, V_n\}$  से कितने अनिर्देशित (आवश्यक नहीं है कि जुड़े हुए हैं) ग्राफ बनाए जा सकते हैं?
- (a)  $n(n-1)/2$  (b)  $2^n$   
 (c)  $n!$  (d)  $2n(n-1)/2$
122. What is the maximum number of edges in an acyclic undirected graph with n vertices?/n शीर्षों वाले acyclic अनिर्देशित ग्राफ में अधिकतम किनारों की संख्या कितनी होगी?
- (a)  $n + 1$  (b)  $n$   
 (c)  $n + 1$  (d)  $2n + 1$
123. What is the largest integer m such that every simple connected graph with n vertices and n edges contains at least m different spanning trees?/n शीर्ष और n किनारों वाले simple connected graph में कम से कम कितने spanning trees होंगे?
- (a) 1 (b) 2  
 (c) 3 (d)  $n$
124. Which of the following data structure is useful in traversing a given graph by breadth first search?/ Breadth First Search (BFS) में कौन सी डेटा संरचना उपयोग होती है?
- (a) Stack (b) List  
 (c) Queue (d) None of the above
125. Array implementation of Stack is not dynamic, which of the following statements supports this argument?/Stack के Array implementation को dynamic नहीं माना जाता। निम्नलिखित में से कौन-सा कथन इस तर्क का समर्थन करता है?
- (a) space allocation for array is fixed and cannot be changed during run-time  
 (b) user unable to give the input for stack operations

- (c) a runtime exception halts execution  
(d) improper program compilation
126. Number of comparisons required for an unsuccessful search of an element in a sequential search, in an ordered list of length  $L$  is/लंबाई  $L$  की क्रमबद्ध सूची में असफल sequential search के लिए तुलना की संख्या कितनी होगी?  
(a)  $L$  (b)  $L/2$   
(c)  $(L + 1)/2$  (d)  $2L$
127. The time taken by binary search algorithm to search a key in a sorted array of  $n$  elements is/ $n$  तत्वों की sorted array में binary search का समय क्या होगा?  
(a)  $O(\log n)$  (b)  $O(n)$   
(c)  $O(n \log n)$  (d)  $O(n^2)$
128. What is the best time complexity of bubble sort?  
Bubble Sort की best case time complexity क्या है?  
(a)  $N^2$  (b)  $N \log N$   
(c)  $N$  (d)  $N (\log N)^2$
129. Which of the following operations is not  $O(1)$  for an array of sorted data (distinct elements)?  
Sorted array (distinct elements) में कौन-सा ऑपरेशन  $O(1)$  नहीं है?  
(a) Find the  $i$ -th largest element  
(b) Delete an element  
(c) Find the  $i$ -th smallest element  
(d) All of the above
130. Which of the following is/are the levels of implementation of data structure?/डेटा संरचना के implementation स्तर कौन-कौन से हैं?  
(a) Abstract level (b) Application level  
(c) Implementation level (d) All of the above
131. Which of the following is not part of ADT description?  
ADT (Abstract Data Type) के विवरण का भाग कौन सा नहीं है?  
(a) Data (b) Operations  
(c) Both of the above (d) None of the above
132. Which of the following data structure is non-linear type?/  
निम्नलिखित में से कौन सी डेटा संरचना non-linear है?  
(a) Strings (b) Lists  
(c) Stacks (d) Graph
133. Which of the following data structure is linear type?  
निम्नलिखित में से कौन सी डेटा संरचना linear है?  
(a) Graph (b) Trees  
(c) Binary tree (d) Stack
134. Match the following:  
(a) Completeness  
(b) Time Complexity  
(c) Space Complexity  
(i) How long does it take to find a solution  
(ii) How much memory is needed  
(iii) Is the strategy guaranteed to find a solution  
सही मिलान कीजिए  
(a) a-iii, b-i, c-ii (b) a-i, b-ii, c-iii  
(c) a-ii, b-i, c-iii (d) a-iii, b-ii, c-i
135. Which of the following is not the internal sort?/निम्नलिखित में से कौन सा internal sort नहीं है?  
(a) Insertion Sort (b) Bubble Sort  
(c) Merge Sort (d) Heap Sort
136. The property of binary tree is/Binary tree का गुणधर्म क्या है?  
(a) The first subtree is called left subtree  
(b) The second subtree is called right subtree  
(c) The root cannot contain NULL  
(d) The right subtree can be empty
137. State true or false:  
(i) The degree of root node is always zero  
(ii) Nodes that are not root and not leaf are called internal nodes  
सही या गलत बताएं  
(a) True, True (b) True, False  
(c) False, True (d) False, False
138. A \_\_\_\_\_ is an acyclic digraph, which has only one node with indegree 0./\_\_\_\_\_ acyclic digraph है, जिसमें केवल एक नोड का indegree 0 होता है?  
(a) Directed tree (b) Undirected tree  
(c) Disjoint tree (d) Direction oriented tree
139. A \_\_\_\_\_ is a directed tree in which out-degree of each node is less than or equal to two./एक \_\_\_\_\_ directed tree है जिसमें प्रत्येक नोड की out-degree दो या उससे कम होती है?  
(a) Unary tree (b) Binary tree  
(c) Ternary tree (d) Both B and C
140. State true or false:/ सही या गलत बताएं  
(i) An empty tree is also a binary tree / एक empty tree भी binary tree होता है  
(ii) In strictly binary tree, the out-degree of every node is either 0 or 2 / एक strictly binary tree में प्रत्येक नोड की out-degree या तो शून्य होती है या 2 होती है?  
(a) True, False (b) False, True  
(c) True, True (d) False, False
141. The postfix form of the expression  $(A + B) * (C * D - E) * F / G$  is? अभिव्यक्ति  $(A + B) * (C * D - E) * F / G$  का Postfix क्या होगा?  
(a)  $AB + CD * E - FG / **$  (b)  $AB + CD * E - F ** G /$   
(c)  $AB + CD * E - * F * G /$  (d)  $AB + CDE * - * F * G /$
142. The data structure required to check whether an expression contains a balanced parenthesis is? / यह जाँचने के लिए कि किसी अभिव्यक्ति (expression) में संतुलित (balanced parenthesis) हैं या नहीं, किस डेटा संरचना की आवश्यकता होती है?  
(a) Stack (b) Queue  
(c) Array (d) Tree
143. What data structure would you most likely see in non recursive implementation of a recursive algorithm?/किसी पुनरावृत्ति (recursive) एल्गोरिथ्म के non-recursive (अप्रत्यक्ष पुनरावृत्ति रहित) कार्यान्वयन में आप सबसे अधिक किस डेटा संरचना को देखेंगे?  
(a) Linked List (b) Stack  
(c) Queue (d) Tree

```

int* ptr;
int n, i;

printf("Enter number of elements:");
scanf("%d",&n);

printf("Entered number of elements: %d\n", n);

ptr = (int*)malloc(n * sizeof(int));

if (ptr == NULL) {
printf("Memory not allocated.\n");
exit(0);
}
else
{
printf("Memory successfully allocated using malloc.\n");

for (i = 0; i < n; ++i) {
ptr[i] = i + 1;
}

printf("The elements of the array are: ");
for (i = 0; i < n; ++i) {
printf("%d, ", ptr[i]);
}
}

return 0;
}

```

### calloc() Function in C

- calloc() function की मदद से हम memory के कई blocks dynamically allocate कर सकते हैं।
- इसका उपयोग सामान्यतः array और structure जैसे complex data structures की memory allocate करने के लिए किया जाता है।
- calloc() भी malloc() की तरह dynamic memory allocate करता है।
- अंतर यह है कि:
- malloc() single memory block allocate करता है।
- calloc() multiple memory blocks allocate करता है।
- malloc() द्वारा allocate की गई memory में default garbage value होती है।
- calloc() द्वारा allocate की गई memory में default रूप से zero initialize होता है।
- यदि calloc() memory allocate नहीं कर पाता, तो यह NULL return करता है।

#### Syntax

```
ptr = (cast-type*) calloc(n, element-size);
```

- यहाँ n = elements की संख्या को दर्शाता है।
- element-size = प्रत्येक element की size को दर्शाता है।

#### Example

```
ptr = (float*) calloc(25, sizeof(float));
```

- यह statement 25 elements के लिए memory allocate करेगा।

- प्रत्येक element की size float data type के बराबर होगी।

#### Example Program

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
int* ptr;
int n, i;

n = 5;
printf("Enter number of elements: %d\n", n);
```

```
ptr = (int*)calloc(n, sizeof(int));
```

```
if (ptr == NULL) {
printf("Memory not allocated.\n");
exit(0);
}
else {
```

```
printf("Memory successfully allocated using calloc.\n");
for (i = 0; i < n; ++i) {
ptr[i] = i + 1;
}
```

```
printf("The elements of the array are: ");
for (i = 0; i < n; ++i) {
printf("%d, ", ptr[i]);
}
}
```

```
return 0;
}
```

### realloc() Function in C

- realloc() function का उपयोग malloc() या calloc() द्वारा allocate की गई dynamic memory का size बदलने के लिए किया जाता है।
- इसकी मदद से हम पहले से allocate memory को resize (increase या decrease) कर सकते हैं।
- यदि हमें dynamic memory का size बढ़ाना या घटाना हो, तो realloc() function का उपयोग किया जाता है।

#### Syntax

```
ptr = realloc(ptr, new-size);
```

यहाँ ptr pointer को नए size के साथ reallocate किया जाता है।

Output (Example Program)

#### Enter a number 3

Memory successfully allocated using calloc.

Enter number of elements: 3

The elements of the array are: 1, 2, 3

Enter the new size of the array: 10

Memory successfully re-allocated using realloc.

The elements of the array are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

- इस उदाहरण में user से एक number input लिया गया।
- फिर उस number के आधार पर calloc() function द्वारा dynamic memory allocate की गई।
- बाद में realloc() function की सहायता से dynamic memory का size बदलकर 10 कर दिया गया।

### free() Function in C

- free() function का उपयोग malloc() या calloc() द्वारा allocate की गई dynamic memory को deallocate या release करने के लिए किया जाता है।
- free() function का उपयोग करने से memory leak (मेमोरी हानि) से बचा जा सकता है।
- यदि हम free() function का उपयोग नहीं करते, तो allocated memory reserve ही रहती है, चाहे program में उसका उपयोग हो या न हो।

### Syntax

free(ptr);

- यहाँ ptr pointer variable है, जो dynamic memory location को point करता है।
- इस pointer और free() function की सहायता से dynamic memory को release किया जाता है।

### Example Program

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
int n, i, *ptr;

printf("Enter number of elements: ");
scanf("%d", &n);

ptr = (int*) malloc(n * sizeof(int));

if(ptr == NULL) {
printf("Error! memory not allocated.");
exit(0);
}

printf("Enter elements: ");
for(i = 0; i < n; ++i) {
scanf("%d", ptr + i);
}

free(ptr);

return 0;
}
```

### File Handling

#### File क्या होती है

- File bytes का एक संग्रह (collection) होती है, जो secondary storage device जैसे disk में store होती है।
- किसी भी system में मुख्यतः दो प्रकार की files होती हैं।
  - Text file
  - Binary file

### Text File

- Text files सामान्यतः .txt extension वाली files होती हैं।
- इन्हें बनाना और उपयोग करना काफी आसान होता है।
- इन files को किसी भी text editor जैसे Notepad की सहायता से बनाया जा सकता है।
- जब text file को open किया जाता है, तो उसका पूरा content plain text के रूप में दिखाई देता है।
- इन files के content को आसानी से edit या delete किया जा सकता है।

### Binary File

- कंप्यूटर में .bin extension वाली files को binary files कहा जाता है।
- Binary files में information plain text के रूप में नहीं बल्कि binary numbers (0 और 1) के रूप में store होती है।
- Binary files में बड़ी मात्रा में data store किया जा सकता है।
- क्योंकि इसमें data binary format (0 और 1) में होता है, इसलिए इन्हें आसानी से पढ़ा नहीं जा सकता।
- Binary files सामान्यतः human readable नहीं होती हैं।

### File Handling in C

- C language में किसी भी variable को memory तभी मिलती है जब program RAM में load होता है।
- RAM एक volatile memory है।
- इसलिए program RAM से बाहर जाते ही उसका सारा data नष्ट हो जाता है।
- जब program RAM से हटता है, तब variables को allocated memory भी deallocate हो जाती है।
- इसलिए variables में stored values को permanent रूप से store नहीं किया जा सकता।
- variable की life program की life के बराबर होती है।
- जैसे ही program समाप्त होता है, variable भी समाप्त हो जाते हैं।
- यदि हमें variables की values program समाप्त होने के बाद भी सुरक्षित रखनी हों, तो उन्हें permanent storage (जैसे hard disk) में file के रूप में store करना पड़ता है।
- इसके लिए program में ऐसा code लिखा जाता है जिससे variables का data secondary storage में file के अंदर save हो जाए।
- इस प्रक्रिया को File Handling कहा जाता है।
- File handling एक तकनीक है जिसके माध्यम से program के data को disk files में permanently store किया जा सकता है।
- File handling की मदद से हम variables में stored data को files में save कर सकते हैं और बाद में उसे पुनः program में उपयोग कर सकते हैं।
- C language में file handling द्वारा हम files को create, update, read और delete कर सकते हैं।

### File Handling की Categories

- File handling को मुख्यतः दो categories में विभाजित किया जाता है।
- High Level File Handling (Standard files / Stream oriented files)
- Low Level File Handling (System oriented files)
- High level file handling को library functions द्वारा manage किया जाता है।
- Low level file handling को system calls द्वारा manage किया जाता है।

- सामान्यतः high level file handling का उपयोग अधिक किया जाता है, क्योंकि इसे manage करना आसान होता है।
- File handling करने के लिए program में कई file functions और operations का उपयोग किया जाता है।
- आगे हम इन file functions और operations के बारे में विस्तार से अध्ययन करेंगे।

### Functions Use In File Handling

No.	Function	Description
1.	fopen()	इस function का कार्य file को open करना होता है।
2.	fprintf()	इस फंक्शन के द्वारा हम file में data को write कर सकते हैं।
3.	fscanf()	इसके द्वारा file में से data को read किया जाता है।
4.	fputc()	इस function का प्रयोग करके फाइल में एक character को write किया जाता है।
5.	fgetc()	इस function का प्रयोग करके फाइल में एक character को write किया जाता है।
6.	fgetc()	यह file में से एक character को read करता है।
7.	fclose()	इस फंक्शन के द्वारा फाइल को close किया जाता है।
8.	fseek()	यह function का file में integer को set करता है।
9.	fputw()	यह function file में integer को write करता है।
9.	fgetw()	यह file में से integer को read करता है।
10.	ftell()	यह current position को return करता है।
11.	rewind()	यह file की शुरुआत में file pointer को set करता है।
12.	ftell()	यह file को पुनः शुरुआत से read करता है।
13.	ftell()	यह file की शुरुआत में file pointer को set करता है।
11.	rewind()	यह file की शुरुआत में file pointer को set करता है।

### File Handling Functions

- ये सभी functions file handling में उपयोग किए जाते हैं, जिनका प्रयोग हम program बनाते समय files को manage करने के लिए करते हैं।

### File Operations in C

C language में हम निम्न प्रकार के file operations perform कर सकते हैं:

- Creation of a new file – नई फाइल बनाना।
- Opening an existing file – पहले से बनी हुई फाइल को खोलना।
- Reading from file – फाइल से data पढ़ना।
- Writing to a file – फाइल में data लिखना।
- Moving to a specific location in a file – फाइल के किसी विशेष स्थान पर जाना।
- Closing a file – फाइल को बंद करना।

### Opening or Creation of a New File

- C language में नई फाइल बनाने या पहले से बनी फाइल को खोलने के लिए fopen() function का उपयोग किया जाता है।
- fopen() function stdio.h header file में defined होता है।

### Syntax

FILE \*fp;

fp = fopen("file\_name", "mode");

- यहाँ fp एक FILE pointer है, जो fopen() द्वारा open की गई file को point करता है।
- FILE एक structure होता है, जिसमें file से संबंधित कई प्रकार की information stored रहती है, जैसे:
  - File का name
  - File का size
  - Buffer size
  - Current position
  - End of file (EOF) आदि।

### Syntax Explanation

- file\_name – यहाँ उस file का नाम दिया जाता है जिसे हम open करना चाहते हैं।
- mode – यहाँ यह निर्धारित किया जाता है कि file पर कौन सा operation करना है, जैसे:
  - Read
  - Write
  - Update

### Example

FILE \*fp;

fp = fopen("fileName.txt", "w");

- इस उदाहरण में fileName.txt उस file का नाम है जिसे हम open करना चाहते हैं।
- यहाँ "w" file opening mode है, जो यह बताता है कि file को write operation के लिए open किया गया है।

### File Opening Modes

- C language में अलग-अलग परिस्थितियों के अनुसार कई file opening modes उपलब्ध होते हैं।
- प्रत्येक mode का अपना अलग उद्देश्य और उपयोग होता है।
- इन modes की सहायता से हम file में read, write या update operations perform कर सकते हैं।
- आगे हम इन सभी file opening modes के बारे में विस्तार से जानेंगे।

Mode	Description
r	यह text फाइल को read mode में open करता है।
w	यह text फाइल को write mode में open करता है।
a	यह text फाइल को append mode में open करता है।
r+	यह text फाइल को read और write दोनों modes में open करता है।
w+	यह text फाइल को read और write दोनों modes में open करता है।
a+	यह text फाइल को read और write दोनों modes में open करता है।
rb	यह binary file को read mode में open करता है।
wb	यह binary file को write mode में open करता है।
ab	यह binary file को append mode में open करता है।
rb+	यह binary file को read mode में open करता है।
wb	यह binary file को write mode में open करता है।
ab	यह binary file को append mode में open करता है।
rb+	यह binary file को read और write दोनों modes में open करता है।
wb+	यह binary file को read और write दोनों modes में open करता है।
ab+	यह binary file को read और write दोनों modes में open करता है।
ab+	यह binary file को read और write दोनों modes में open करता है।

### How fopen() Function Works

- जब भी हम fopen() function को call करते हैं, तो यह सबसे पहले उस file को computer की disk में खोजता है, जिसका नाम हमने fopen() के parentheses में दिया होता है।
- यदि fopen() को वह file disk में मिल जाती है, तो वह उस file को disk से buffer space में load करता है।
- इसके बाद यह file को buffer से required mode (जैसे read, write आदि) में open कर देता है।
- यदि fopen() को वह file disk में नहीं मिलती, तो कुछ modes में यह एक नई file create कर देता है और उसे open कर देता है।