Unit 1 : Basic Concept and Introduction to Data structure

1. Which of these best describes an array?
   a) A data structure that shows a hierarchical behaviour
   b) Container of objects of similar types
   c) Arrays are immutable once initialised
   d) Array is not a data structure

2. How do you initialize an array ?
   a) int arr[3] = (1,2,3);
   b) int arr(3) = {1,2,3};
   c) int arr[3] = {1,2,3};
   d) int arr(3) = (1,2,3);

3. When does the ArrayIndexOutOfBoundsException occur?
   a) Compile-time
   b) Run-time
   c) Not an error
   d) Not an exception at all

4. Which of the following concepts make extensive use of arrays?
   a) Binary trees
   b) Scheduling of processes
   c) Caching
   d) Spatial locality

5. What are the advantages of arrays?
   a) Objects of mixed data types can be stored
   b) Elements in an array cannot be sorted
   c) Index of first element of an array is 1
   d) Easier to store elements of same data type

6. What are the disadvantages of arrays?
   a) Data structure like queue or stack cannot be implemented
   b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
   c) Index value of an array can be negative
   d) Elements are sequentially accessed

7. Assuming int is of 4bytes, what is the size of int arr[15];?
   a) 15
   b) 19
   c) 11
   d) 60

8. In general, the index of the first element in an array is _____
   a) 0
   b) -1
   c) 2
   d) 1

9. Elements in an array are accessed _____
   a) randomly
   b) sequentially
   c) exponentially
   d) logarithmically

Unit : Stack and Queue

10. Process of inserting an element in stack is called _____
    a) Create
    b) Push
    c) Evaluation
    d) Pop

11. Process of removing an element from stack is called _____
    a) Create
    b) Push
    c) Evaluation
    d) Pop

12. In a stack, if a user tries to remove an element from empty stack it is called _____
    a) Underflow
    b) Empty collection
    c) Overflow
    d) Garbage Collection

13. Pushing an element into stack already having five elements and stack size of 5, then stack becomes
    a) Overflow
    b) Crash
    c) Underflow
    d) User flow

14. 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
15. . Which of the following applications may use a 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

16. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (()(())(())) are:
    a) 1
    b) 2
    c) 3

d) 4 or more

17. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence that contains 2 left parentheses and 3 right parentheses (in some order).
The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation?
a) 1
b) 2
c) 3
d) 4 or more

18. What is the value of the postfix expression 6 3 2 4 + – *?
a) 1
b) 40
c) 74
d) -18

19. Here is an infix expression: 4 + 3*(6*3-12). Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation.
The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?
a) 1
b) 2
c) 3
d) 4

20. 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 /

21. The data structure required to check whether an expression contains balanced parenthesis is?
a) Stack
b) Queue
c) Array
d) Tree

22. What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm?
a) Linked List
b) Stack

c) Queue
d) Tree

23. The process of accessing data stored in a serial access memory is similar to manipulating data on a _____
a) Heap
b) Binary Tree
c) Array
d) Stack

24. The postfix form of A*B+C/D is?
a) *AB/CD+
b) AB*CD/+
c) A*BC+/D
d) ABCD+/*

25. Which data structure is needed to convert infix notation to postfix notation?
a) Branch
b) Tree
c) Queue
d) Stack

26. The prefix form of A-B/ (C * D ^ E) is?
a) -/*^ACBDE
b) -ABCD*^DE
c) -A/B*C^DE
d) -A/BC*^DE

27. What is the result of the following operation?
Top (Push (S, X))
a) X
b) X+S
c) S
d) XS

28. The prefix form of an infix expression (p + q) – (r * t) is?
a) + pq – *rt
b) – +pqr * t
c) – +pq * rt
d) – + * pqrt

29. . Which data structure is used for implementing recursion?
    a) Queue
    b) Stack
    c) Array
    d) List

30. The result of evaluating the postfix expression 5, 4, 6, +, *, 4, 9, 3, /, +, * is?
    a) 600
    b) 350
    c) 650
    d) 588

31. Convert the following infix expressions into its equivalent postfix expressions
    (A + B ⋀D)/(E – F)+G
    a) (A B D ⋀ + E F – / G +)
    b) (A B D +⋀ E F – / G +)
    c) (A B D ⋀ + E F/- G +)
    d) (A B D E F + ⋀ / – G +)

32. Convert the following Infix expression to Postfix form using a stack
    x + y * z + (p * q + r) * s, Follow usual precedence rule and assume that the expression is legal.
    a) xyz*+pq*r+s*+
    b) xyz*+pq*r+s+*
    c) xyz+*pq*r+s*+
    d) xyzp+**qr+s*+

33. Which of the following statement(s) about stack data structure is/are NOT correct?
    a) Linked List are used for implementing Stacks
    b) Top of the Stack always contain the new node
    c) Stack is the FIFO data structure
    d) Null link is present in the last node at the bottom of the stack

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 are

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. Assume that the operators +,-, X are left associative and ^ is right associative.
The order of precedence (from highest to lowest) is ^, X, +, -. The postfix expression for the infix expression a + b X c – d ^ e ^ f is
a) abc X+ def ^^ –
b) abc X+ de^f^ –
c) ab+c Xd – e ^f^
d) -+aXbc^ ^def

38. 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

39. 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 ?
a) Queue
b) Stack
c) Tree
d) Linked list

40. The data structure required for Breadth First Traversal on a graph is?
a) Stack
b) Array
c) Queue

d) Tree

41. A queue follows _____
    a) FIFO (First In First Out) principle
    b) LIFO (Last In First Out) principle
    c) Ordered array
    d) Linear tree

42. Circular Queue is also known as _____
    a) Ring Buffer
    b) Square Buffer
    c) Rectangle Buffer
    d) Curve Buffer

43. 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

44. A data structure in which elements can be inserted or deleted at/from both the ends but not in the middle is?
    a) Queue
    b) Circular queue
    c) Dequeue
    d) Priority queue

45. A normal queue, if implemented using an array of size MAX_SIZE, gets full when
    a) Rear = MAX_SIZE – 1
    b) Front = (rear + 1)mod MAX_SIZE
    c) Front = rear + 1
    d) Rear = front

46. 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

47. Which of the following is not the type of queue?
    a) Ordinary queue
    b) Single ended queue

c) Circular queue
d) Priority queue

Unit : Linked List

48. 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

49. 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

50. In linked list each node contain 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

51. 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) $\theta(1)$

52. 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) $O(n^3)$

53. What would be the asymptotic time complexity to find an element in the linked list?
    a) O(1)
    b) O(n)
    c) $O(n^2)$
    d) $O(n^4)$

54. What would be the asymptotic time complexity to insert an element at the second position in the linked list?
    a) O(1)
    b) O(n)
    c) $O(n^2)$
    d) $O(n^3)$

55. The concatenation of two list can performed in O(1) time. Which of the following variation of linked list can be used?
    a) Singly linked list
    b) Doubly linked list
    c) Circular doubly linked list
    d) Array implementation of list

56. What kind of linked list is best to answer question like "What is the item at position n?"
    a) Singly linked list
    b) Doubly linked list
    c) Circular linked list
    d) Array implementation of linked list

57. Linked lists are not suitable to for the implementation of?
    a) Insertion sort
    b) Radix sort
    c) Polynomial manipulation
    d) Binary search

58. Linked list is considered as an example of _____ type of memory allocation.
    a) Dynamic
    b) Static
    c) Compile time
    d) Heap

59. In Linked List implementation, a node carries information regarding _____
    a) Data
    b) Link

c) Data and Link
d) Node

60. 5. Linked list data structure offers considerable saving in _____
a) Computational Time
b) Space Utilization
c) Space Utilization and Computational Time
d) Speed Utilization

61. Which of the following points is/are not true about Linked List data structure when it is compared with array?
a) Arrays have better cache locality that can make them better in terms of performance
b) It is easy to insert and delete elements in Linked List
c) Random access is not allowed in a typical implementation of Linked Lists
d) Access of elements in linked list takes less time than compared to arrays

62. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?
a) Insertion Sort
b) Quick Sort
c) Heap Sort
d) Merge Sort

63. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is
a) log 2 n
b) $^{n}/_{2}$
c) log 2 n – 1
d) n

64. 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 the node X from given linked list?
a) Possible if X is not last node
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

65. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?
a) Delete the first element
b) Insert a new element as a first element
c) Delete the last element of the list

d) Add a new element at the end of the list

66. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is
    a) log2 n
    b) $^{n}/_{2}$
    c) log2 n – 1
    d) n

67. Which of the following is not a disadvantage to the usage of array?
    a) Fixed size
    b) 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

68. What is the time complexity of inserting at the end in dynamic arrays?
    a) O(1)
    b) O(n)
    c) O(logn)
    d) Either O(1) or O(n)

69. What is the time complexity to count the number of elements in the linked list?
    a) O(1)
    b) O(n)
    c) O(logn)
    d) $O(n^2)$

70. What is the space complexity for deleting a linked list?
    a) O(1)
    b) O(n)
    c) Either O(1) or O(n)
    d) O(logn)

71. 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

72. 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

c) 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

73. How do you calculate the pointer difference in a memory efficient double linked list?
a) head xor tail
b) pointer to previous node xor pointer to next node
c) pointer to previous node – pointer to next node
d) pointer to next node – pointer to previous node

74. 6. What is the worst case time complexity of inserting a node in a doubly linked list?
a) O(nlogn)
b) O(logn)
c) O(n)
d) O(1)

75. 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 point to null in a circular linked list
d) Head node is known in circular linked list

76. What is the time complexity of searching for an element in a circular linked list?
a) O(n)
b) O(nlogn)
c) O(1)
d) $O(n^2)$

77. Which of the following application makes use of a circular linked list?
a) Undo operation in a text editor
b) Recursive function calls
c) Allocating CPU to resources
d) Implement Hash Tables

78. 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

79. Consider a small circular linked list. How to detect the presence of cycles in this list effectively?
a) Keep one node as head and traverse another temp node till the end to check if its 'next points to head

b) Have fast and slow pointers with the fast pointer advancing two nodes at a time and slow pointer advancing by one node at a time
c) Cannot determine, you have to pre-define if the list contains cycles
d) Circular linked list itself represents a cycle. So no new cycles cannot be generated

80. 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

81. What does 'stack underflow' refer to?
a) accessing item from an undefined stack
b) adding items to a full stack
c) removing items from an empty stack
d) index out of bounds exception

82. What is the time complexity of pop() operation when the stack is implemented using an array?
a) O(1)
b) O(n)
c) O(logn)
d) O(nlogn)

83. 6. Which of the following array position will be occupied by a new element being pushed for a stack of size N elements(capacity of stack > N).
a) S[N-1]
b) S[N]
c) S[1]
d) S[0]

84. What happens when you pop from an empty stack while implementing using the Stack ADT in Java?
a) Undefined error
b) Compiler displays a warning
c) EmptyStackException is thrown
d) NoStackException is thrown

85. Array implementation of Stack is not dynamic, which of the following statements supports this argument?
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

86. 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).
    a) S[N-1]
    b) S[N]
    c) S[N-2]
    d) S[N+1]

87. What is the best case time complexity of deleting a node in Singly Linked list?
    a) O (n)
    b) O (n^2)
    c) O (nlogn)
    d) O (1)

88. Which of the following statements are not correct with respect to Singly Linked List(SLL) and Doubly Linked List(DLL)?
    a) Complexity of Insertion and Deletion at known position is O(n) in SLL and O(1) in DLL
    b) SLL uses lesser memory per node than DLL
    c) DLL has more searching power than SLL
    d) Number of node fields in SLL is more than DLL

89. What does ‘stack overflow’ refer to?
    a) accessing item from an undefined stack
    b) adding items to a full stack
    c) removing items from an empty stack
    d) index out of bounds exception

90. Which of the following data structures can be used for parentheses matching?
    a) n-ary tree
    b) queue
    c) priority queue
    d) stack

91. 10. Minimum number of queues to implement stack is _____
    a) 3
    b) 4
    c) 1
    d) 2

92. Which of the following properties is associated with a queue?
    a) First In Last Out
    b) First In First Out
    c) Last In First Out

d) Last In Last Out

93. In a circular queue, how do you increment the rear end of the queue?
    a) rear++
    b) (rear+1) % CAPACITY
    c) (rear % CAPACITY)+1
    d) rear–

94. What is the term for inserting into a full queue known as?
    a) overflow
    b) underflow
    c) null pointer exception
    d) program won't be compiled

95. What is the time complexity of enqueue operation?
    a) O(logn)
    b) O(nlogn)
    c) O(n)
    d) O(1)

96. What is the need for a circular queue?
    a) effective usage of memory
    b) easier computations
    c) to delete elements based on priority
    d) implement LIFO principle in queues

97. What is the space complexity of a linear queue having n elements?
    a) O(n)
    b) O(nlogn)
    c) O(logn)
    d) O(1)

98. 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

99. In linked list implementation of a queue, where does a new element be inserted?
    a) At the head of link list
    b) At the centre position in the link list
    c) At the tail of the link list

d) At any position in the linked list

100. In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into a NONEMPTY queue?
a) Only front pointer
b) Only rear pointer
c) Both front and rear pointer
d) No pointer will be changed

101. In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue?
a) Only front pointer
b) Only rear pointer
c) Both front and rear pointer
d) No pointer will be changed

102. In case of insertion into a linked queue, a node borrowed from the _____ list is inserted in the queue.
a) AVAIL
b) FRONT
c) REAR
d) NULL

103. In linked list implementation of a queue, from where is the item deleted?
a) At the head of link list
b) At the centre position in the link list
c) At the tail of the link list
d) Node before the tail

104. In linked list implementation of a queue, the important condition for a queue to be empty is?
a) FRONT is null
b) REAR is null
c) LINK is empty
d) FRONT==REAR-1

105. The essential condition which is checked before insertion in a linked queue is?
a) Underflow
b) Overflow
c) Front value
d) Rear value

106. The essential condition which is checked before deletion in a linked queue is?
a) Underflow
b) Overflow

c) Front value
d) Rear value

107. Which of the following is true about linked list implementation of queue?
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 beginning, then in pop operation, nodes must be removed from the beginning
c) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from end
d) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from beginning

108. With what data structure can a priority queue be implemented?
a) Array
b) List
c) Heap
d) Tree

109. Which of the following is not an application of priority queue?
a) Huffman codes
b) Interrupt handling in operating system
c) Undo operation in text editors
d) Bayesian spam filter

110. What is the time complexity to insert a node based on key in a priority queue?
a) O(nlogn)
b) O(logn)
c) O(n)
d) $O(n^2)$

111. 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

112. Which of the following is not an advantage of priority queue?
a) Easy to implement
b) Processes with different priority can be efficiently handled
c) Applications with differing requirements
d) Easy to delete elements in any case

113. What is the time complexity to insert a node based on position in a priority queue?
    a) O(nlogn)
    b) O(logn)
    c) O(n)
    d) $O(n^2)$

114. What is a dequeue?
    a) A queue with insert/delete defined for both front and rear ends of the queue
    b) A queue implemented with a doubly linked list
    c) A queue implemented with both singly and doubly linked lists
    d) A queue with insert/delete defined for front side of the queue

115. What are the applications of dequeue?
    a) A-Steal job scheduling algorithm
    b) Can be used as both stack and queue
    c) To find the maximum of all sub array
    d) To avoid collision in hash tables

116. What is the time complexity of deleting from the rear end of the dequeue implemented with a singly linked list?
    a) O(nlogn)
    b) O(logn)
    c) O(n)
    d) $O(n^2)$

117. A Double-ended queue supports operations such as adding and removing items from both the sides of the queue. They support four operations like addFront(adding item to top of the queue), addRear(adding item to the bottom of the queue), removeFront(removing item from the top of the queue) and removeRear(removing item from the bottom of the queue). You are given only stacks to implement this data structure. You can implement only push and pop operations. What are the total number of stacks required for this operation?(you can reuse the stack)
    a) 1
    b) 2
    c) 3
    d) 4

118. You are asked to perform a queue operation using a stack. Assume the size of the stack is some value ‘n’ and there are ‘m’ number of variables in this stack. The time complexity of performing deQueue operation is (Using only stack operations like push and pop)(Tightly bound).
    a) O(m)
    b) O(n)
    c) O(m*n)

d) Data is insufficient

119. Consider you have an array of some random size. You need to perform dequeue operation. You can perform it using stack operation (push and pop) or using queue operations itself (enQueue and Dequeue). The output is guaranteed to be same. Find some differences?
a) They will have different time complexities
b) The memory used will not be different
c) There are chances that output might be different
d) No differences

120. Consider you have a stack whose elements in it are as follows.
5 4 3 2 << top
Where the top element is 2.
You need to get the following stack
6 5 4 3 2 << top
The operations that needed to be performed are (You can perform only push and pop):
a) Push(pop()), push(6), push(pop())
b) Push(pop()), push(6)
c) Push(pop()), push(pop()), push(6)
d) Push(6)

121. A double-ended queue supports operations like adding and removing items from both the sides of the queue. They support four operations like addFront(adding item to top of the queue), addRear(adding item to the bottom of the queue), removeFront(removing item from the top of the queue) and removeRear(removing item from the bottom of the queue). You are given only stacks to implement this data structure. You can implement only push and pop operations. What's the time complexity of performing addFront and addRear? (Assume 'm' to be the size of the stack and 'n' to be the number of elements)
a) O(m) and O(n)
b) O(1) and O(n)
c) O(n) and O(1)
d) O(n) and O(m)

122. Why is implementation of stack operations on queues not feasible for a large dataset (Asssume the number of elements in the stack to be n)?
a) Because of its time complexity O(n)
b) Because of its time complexity O(log(n))
c) Extra memory is not required
d) There are no problems

123. Consider yourself to be in a planet where the computational power of chips to be slow. You have an array of size 10.You want to perform enqueue some element into this array. But you can perform only push and pop operations .Push and pop operation both take 1 sec respectively. The total time required to perform enQueue operation is?

a) 20
b) 40
c) 42
d) 43

124. You have two jars, one jar which has 10 rings and the other has none. They are placed one above the other. You want to remove the last ring in the jar. And the second jar is weak and cannot be used to store rings for a long time.
a) Empty the first jar by removing it one by one from the first jar and placing it into the second jar
b) Empty the first jar by removing it one by one from the first jar and placing it into the second jar and empty the second jar by placing all the rings into the first jar one by one
c) There exists no possible way to do this
d) Break the jar and remove the last one

125. Given only a single array of size 10 and no other memory is available. Which of the following operation is not feasible to implement (Given only push and pop operation)?
a) Push
b) Pop
c) Enqueue
d) Returntop

126. Given an array of size n, let's assume an element is 'touched' if and only if some operation is performed on it(for example, for performing a pop operation the top element is 'touched'). Now you need to perform Dequeue operation. Each element in the array is touched atleast?
a) Once
b) Twice
c) Thrice
d) Four times

127. To implement a stack using queue(with only enqueue and dequeue operations), how many queues will you need?
a) 1
b) 2
c) 3
d) 4

128. Express -15 as a 6-bit signed binary number.
a) 001111
b) 101111
c) 101110
d) 001110

129. Which is the predefined method available in Java to convert decimal to binary numbers?
a) toBinaryInteger(int)
b) toBinaryValue(int)
c) toBinaryNumber(int)
d) toBinaryString(int)

130. What is the time complexity for converting decimal to binary numbers?
a) O(1)
b) O(n)
c) O(logn)
d) O(nlogn)

131. How many stacks are required for applying evaluation of infix expression algorithm?
a) one
b) two
c) three
d) four

132. How many passes does the evaluation of infix expression algorithm makes through the input?
a) One
b) Two
c) Three
d) Four

133. Identify the infix expression from the list of options given below.
a) a/b+(c-d)
b) abc*+d+ab+cd+*ce-f-
c) ab-c-
d) +ab

134. Which of the following statement is incorrect with respect to evaluation of infix expression algorithm?
a) Operand is pushed on to the stack
b) If the precedence of operator is higher, pop two operands and evaluate
c) If the precedence of operator is lower, pop two operands and evaluate
d) The result is pushed on to the operand stack

135. Evaluate the following statement using infix evaluation algorithm and choose the correct answer.
1+2*3-2
a) 3
b) 6
c) 5

d) 4

136. Evaluation of infix expression is done based on precedence of operators.
a) True
b) False

137. Of the following choices, which operator has the lowest precedence?
a) ^
b) +
c) /
d) #

138. The system throws an error if parentheses are encountered in an infix expression evaluation algorithm.
a) True
b) False

139. Evaluate the following and choose the correct answer.
a/b+c*d where a=4, b=2, c=2, d=1.
a) 1
b) 4
c) 5
d) 2

140. Evaluate the following statement using infix evaluation algorithm and choose the correct answer.
4*2+3-5/5
a) 10
b) 11
c) 16
d) 12

141. Using the evaluation of infix expression, evaluate a^b+c and choose the correct answer. (a=2, b=2, c=2)
a) 12
b) 8
c) 10
d) 6

142. Evaluate the following infix expression using algorithm and choose the correct answer. a+b*c-d/e^f where a=1, b=2, c=3, d=4, e=2, f=2.
a) 6
b) 8
c) 9
d) 7

143. How many stacks are required for evaluation of prefix expression?
    a) one
    b) two
    c) three
    d) four

144. 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

145. The associativity of an exponentiation operator ^ is right side.
    a) True
    b) False

146. How many types of input characters are accepted by this algorithm?
    a) one
    b) two
    c) three
    d) four

147. 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

148. Find the output of the following prefix expression
    *+2-2 1/-4 2+-5 3 1
    a) 2
    b) 12
    c) 10
    d) 4

149. An error is thrown if the character '\n' is pushed in to the character stack.
    a) true
    b) false

150. Using the evaluation of prefix algorithm, evaluate +-9 2 7.
    a) 10
    b) 4
    c) 17

d) 14

151. If -*+abcd = 11, find a, b, c, d using evaluation of prefix algorithm.
a) a=2, b=3, c=5, d=4
b) a=1, b=2, c=5, d=4
c) a=5, b=4, c=7,d=5
d) a=1, b=2, c=3, d=4

152. What is the other name for a postfix expression?
a) Normal polish Notation
b) Reverse polish Notation
c) Warsaw notation
d) Infix notation

153. Which of the following is an example for a postfix expression?
a) a*b(c+d)
b) abc*+de-+
c) +ab
d) a+b-c

154. Reverse Polish Notation is the reverse of a Polish Notation
a) True
b) False

155. What is the time complexity of evaluation of postfix expression algorithm?
a) O (N)
b) O (N log N)
c) O (N^2)
d) O (M log N)

156. In Postfix expressions, the operators come after the operands.
a) True
b) False

157. Which of these operators have the highest order of precedence?
a) '(' and ')'
b) '*' and '/'
c) '~' and '^'
d) '+' and '-'

158. Which of the following is not an application of stack?
a) evaluation of postfix expression
b) conversion of infix to postfix expression
c) balancing symbols

d) line at ticket counter

159. While evaluating a postfix expression, when an operator is encountered, what is the correct operation to be performed?
a) push it directly on to the stack
b) pop 2 operands, evaluate them and push the result on to the stack
c) pop the entire stack
d) ignore the operator

160. Which of the following statement is incorrect?
a) Postfix operators use value to their right
b) Postfix operators use value to their left
c) Prefix operators use value to their right
d) In postfix expression, operands are followed by operators

161. What is the result of the given postfix expression? abc*+ where a=1, b=2, c=3.
a) 4
b) 5
c) 6
d) 7

162. What is the result of the following postfix expression?
ab*cd*+ where a=2,b=2,c=3,d=4.
a) 16
b) 12
c) 14
d) 10

163. Consider the stack
| 5 |
| 4 |
| 3 |
| 2 |.
At this point, '*' is encountered. What has to be done?
a) 5*4=20 is pushed into the stack
b) * is pushed into the stack
c) 2*3=6 is pushed into the stack
d) * is ignored

164. Evaluate the postfix expression ab + cd/- where a=5, b=4, c=9, d=3.
a) 23
b) 15
c) 6

d) 10

165. Evaluate and write the result for the following postfix expression
abc*+de*f+g*+ where a=1, b=2, c=3, d=4, e=5, f=6, g=2.
a) 61
b) 59
c) 60
d) 55

166. What data structure is used when converting an infix notation to prefix notation?
a) Stack
b) Queue
c) B-Trees
d) Linked-list

167. What would be the Prefix notation for the given equation?

A+(B*C)

a) +A*CB
b) *B+AC
c) +A*BC
d) *A+CB

168. What would be the Prefix notation for the given equation?
(A*B)+(C*D)
a) +*AB*CD
b) *+AB*CD
c) **AB+CD
d) +*BA*CD

169. What would be the Prefix notation for the given equation?
A+B*C^D
a) +A*B^CD
b) +A^B*CD
c) *A+B^CD
d) ^A*B+CD

170. Out of the following operators (^, *, +, &, $), the one having highest priority is _____
a) +
b) $
c) ^

d) &

171. Out of the following operators (|, *, +, &, $), the one having lowest priority is _____
a) +
b) $
c) |
d) &

172. What would be the Prefix notation for the given equation?
A^B^C^D
a) ^^^ABCD
b) ^A^B^CD
c) ABCD^^^
d) AB^C^D

173. What would be the Prefix notation for the given equation?
a+b-c/d&e|f
a) |&-+ab/cdef
b) &|-+ab/cdef
c) |&-ab+/cdef
d) |&-+/abcdef

174. What would be the Prefix notation for the given equation?
(a+(b/c)*(d^e)-f)
a) -+a*/^bcdef
b) -+a*/bc^def
c) -+a*b/c^def
d) -a+*/bc^def

175. What would be the Prefix notation and Postfix notation for the given equation?
A+B+C
a) ++ABC and AB+C+
b) AB+C+ and ++ABC
c) ABC++ and AB+C+
d) ABC+ and ABC+

176. What would be the Prefix notation for the given equation?
a|b&c
a|&bc
b) &|abc
c) |a&bc
d) ab&|c

177. When an operand is read, which of the following is done?
a) It is placed on to the output
b) It is placed in operator stack
c) It is ignored
d) Operator stack is emptied

178. What should be done when a left parenthesis '(' is encountered?
a) It is ignored
b) It is placed in the output
c) It is placed in the operator stack
d) The contents of the operator stack is emptied

179. Which of the following is an infix expression?
a) (a+b)*(c+d)
b) ab+c*
c) +ab
d) abc+*

180. What is the time complexity of an infix to postfix conversion algorithm?
a) O(N log N)
b) O(N)
c) $O(N^2)$
d) O(M log N)

181. What is the postfix expression for the corresponding infix expression?
a+b*c+(d*e)
a) abc*+de*+
b) abc+*de*+
c) a+bc*de+*
d) abc*+(de)*+

182. Parentheses are simply ignored in the conversion of infix to postfix expression.
a) True
b) False

183. It is easier for a computer to process a postfix expression than an infix expression.
a) True
b) False

184. What is the postfix expression for the infix expression?
a-b-c
a) -ab-c
b) ab – c –
c) – -abc

d) -ab-c

185. What is the postfix expression for the following infix expression?
a/b^c-d
a) abc^/d-
b) ab/cd^-
c) ab/^cd-
d) abcd^/-

186. Which of the following statement is incorrect with respect to infix to postfix conversion algorithm?
a) operand is always placed in the output
b) operator is placed in the stack when the stack operator has lower precedence
c) parenthesis are included in the output
d) higher and equal priority operators follow the same condition

187. In infix to postfix conversion algorithm, the operators are associated from?
a) right to left
b) left to right
c) centre to left
d) centre to right

188. What is the corresponding postfix expression for the given infix expression?
a*(b+c)/d
a) ab*+cd/
b) ab+*cd/
c) abc*+/d
d) abc+*d/

189. What is the corresponding postfix expression for the given infix expression?
a+(b*c(d/e^f)*g)*h)
a) ab*cdef/^*g-h+
b) abcdef^/*g*h*+
c) abcd*^ed/g*-h*+
d) abc*de^fg/*-*h+

190. What is the correct postfix expression for the following expression?
a+b*(c^d-e)^(f+g*h)-i
a) abc^de-fg+*^*+i-
b) abcde^-fg*+*^h*+i-
c) abcd^e-fgh*+^*+i-
d) ab^-dc*+ef^gh*+i-

Unit : Graph

191. 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
192. 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
193. In a simple graph, the number of edges is equal to twice the sum of the degrees of the vertices.
    a) True
    b) False

194. A connected planar graph having 6 vertices, 7 edges contains _____ regions.
    a) 15
    b) 3
    c) 1
    d) 11

195. If a simple graph G, contains n vertices and m edges, the number of edges in the Graph G'(Complement of G) is _____
    a) (n*n-n-2*m)/2
    b) (n*n+n+2*m)/2
    c) (n*n-n-2*m)/2
    d) (n*n-n+2*m)/2

196. Which of the following properties does a simple graph not hold?
    a) Must be connected
    b) Must be unweighted
    c) Must have no loops or multiple edges
    d) Must have no multiple edges

197. What is the maximum number of edges in a bipartite graph having 10 vertices?
    a) 24
    b) 21
    c) 25
    d) 16

198. 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

199. 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
b) v = e+1
c) v + 1 = e
d) v = e-1

200. For which of the following combinations of the degrees of vertices would the connected graph be eulerian?
a) 1,2,3
b) 2,3,4
c) 2,4,5
d) 1,3,5

201. A graph with all vertices having equal degree is known as a _____
a) Multi Graph
b) Regular Graph
c) Simple Graph
d) Complete Graph

202. 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

Unit : Tree

203 How many children does a binary tree have?
a) 2
b) any number of children
c) 0 or 1 or 2
d) 0 or 1

204. 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

205. What must be the ideal size of array if the height of tree is 'l'?
a) 2^l-1
b) l-1
c) l
d) 2l

206 What are the children for node 'w' of a complete-binary tree in an array representation?
a) 2w and 2w+1
b) 2+w and 2-w
c) w+1/2 and w/2
d) w-1/2 and w+1/2

207. What is the parent for a node 'w' of a complete binary tree in an array representation when w is not 0?
a) floor(w-1/2)
b) ceil(w-1/2)
c) w-1/2
d) w/2

208. If the tree is not a complete binary tree then what changes can be made for easy access of children of a node in the array?
a) every node stores data saying which of its children exist in the array
b) no need of any changes continue with 2w and 2w+1, if node is at i
c) keep a seperate table telling children of a node
d) use another array parallel to the array with tree

Unit : Searching and Sorting Techniques

209 Which of the following is not a stable sorting algorithm?
a) Insertion sort
b) Selection sort
c) Bubble sort
d) Merge sort

210. Which of the following is a stable sorting algorithm?
a) Merge sort
b) Typical in-place quick sort
c) Heap sort
d) Selection sort

211. Which of the following is not an in-place sorting algorithm?
a) Selection sort
b) Heap sort
c) Quick sort
d) Merge sort

212. Running merge sort on an array of size n which is already sorted is
a) O(n)
b) O(nlogn)
c) $O(n^2)$
d) None

213. The time complexity of a quick sort algorithm which makes use of median, found by an O(n) algorithm, as pivot element is
a) $O(n^2)$
b) O(nlogn)
c) O(nloglogn)
d) O(n)

214. Which of the following is not a noncomparison sort?
a) Counting sort
b) Bucket sort
c) Radix sort
d) Shell sort

215. The time complexity of heap sort in worst case is
a) O(logn)
b) O(n)
c) O(nlogn)
d) $O(n^2)$

216. If the given input array is sorted or nearly sorted, which of the following algorithm gives the best performance?
a) Insertion sort
b) Selection sort
c) Quick sort
d) Merge sort

217. Which of the following algorithm pays the least attention to the ordering of the elements in the input list?
a) Insertion sort
b) Selection sort
c) Quick sort
d) None

218. Consider the situation in which assignment operation is very costly. Which of the following sorting algorithm should be performed so that the number of assignment operations is minimized in general?
a) Insertion sort
b) Selection sort
c) Heap sort
d) None

219. Time complexity of bubble sort in best case is
a) θ (n)
b) θ (nlogn)
c) θ (n^2)
d) θ $(n(\log n)^2)$

220. Given a number of elements in the range $[0 \ldots n^3]$. which of the following sorting algorithms can sort them in O(n) time?
a) Counting sort
b) Bucket sort
c) Radix sort
d) Quick sort

221. Which of the following algorithms has lowest worst case time complexity?
a) Insertion sort
b) Selection sort

c) Quick sort
d) Heap sort

222. Which of the following sorting algorithms is/are stable
a) Counting sort
b) Bucket sort
c) Radix sort
d) All of the above

223. Counting sort performs ……. Numbers of comparisons between input elements.
a) 0
b) n
c) nlogn
d) n^2

Answers:

1	2	3	4	5	6	7	8	9	10
b	c	b	d	d	b	d	a	a	B
11	12	13	14	15	16	17	18	19	20
d	a	a	d	d	c	b	d	d	C
21	22	23	24	25	26	27	28	29	30
a	b	d	b	d	c	a		b	B
31	32	33	34	35	36	37	38	39	40
a	a	c	a	d	c	b	b	a	C
41	42	43	44	45	46	47	48	49	50
a	a	a	c	a	c	b	a	b	C
51	52	53	54	55	56	57	58	59	60
c	b	b	a	c	d	d	a	b	C
61	62	63	64	65	66	67	68	69	70
d	d	d	a	c	d	d	d	b	A
71	72	73	74	75	76	77	78	79	80
d	a	b	c	c	a	c	b	b	A
81	82	83	84	85	86	87	88	89	90
c	a	b	c	a	a	d	d	b	D
91	92	93	94	95	96	97	98	99	100
c	b	b	a	d	a	a	d	c	B
101	102	103	104	105	106	107	108	109	110
c	a	a	a	b	a	a	d	c	C
111	112	113	114	115	116	117	118	119	120
c	d	c	a	d	c	b	a	a	A
121	122	123	124	125	126	127	128	129	130
b	a	d	b	c	d	b	d	b	C
131	132	133	134	135	136	137	138	139	140
b	a	a	b	c	a	d	b	b	A
141	142	143	144	145	146	147	148	149	150
d	a	b	b	a	c	a	a	b	D
151	152	153	154	155	156	157	158	159	160
b	b	b	b	a	a	c	d	b	A
161	162	163	164	165	166	167	168	169	170
d	a	a	c	b	a	c	a	a	C
171	172	173	174	175	176	177	178	179	180
c	a	a	b	a	c	a	c	a	B
181	182	183	184	185	186	187	188	189	190
a	b	b	b	a	c	b	d	b	C

191	192	193	194	195	196	197	198	199	200
a	b	b	b	a	a	c	b	b	a
201	202	203	204	205	206	207	208	209	210
b	c	c	c	a	a	a	a	b	A
211	212	213	214	215	216	217	218	219	220
d	b	b	d	c	a	b	b	a	C
221	222	223							
d	d	a							