**1.**               **The recurrence relation that arises in relation with the complexity of binary search is**

**a.**        **T(n)=T(n/2)+k, where k is constant**

b.         T(n)=2T(n/2)+k, where k is constant

c. T(n)=T(n/2)+log(n)

d.         T(n)=T(n/2)+n

**2.**               **The running time T(n), where `n' is the input size of a recursive algorithm is given as follows T(n)=c+T(n-1),if n>1  
d, if n≤ 1  
The order of this algorithm is**

a.        n2

**b.**        **n**

c. n3

d.        nn

**3.**               **The concept of order(Big O) is important because**

**a.**        **it can be used to decide the best algorithm that solves a given problem**

b.         It determines the minimum size of a problem that can be solved in a given system, in a given amount of time

c. It is the lower bound of the growth rate of the algorithm

d.         It is the average bound of the growth rate of the algorithm

**4.**               **The concept of order (Big O) is important because**

a.         it can not be used to decide the best algorithm that solves a given problem

**b.**        **It determines the maximum size of a problem that can be solved in a given system, in a given amount of time**

c. It is the lower bound of the growth rate of the algorithm

d.         It is the average bound of the growth rate of the algorithm

**5.**               **The time complexity of an algorithm T(n), where n is the input size is given byT(n)=T(n-1)+/n, if n>1 =1 otherwise  
The order of the algorithm is**

**a.**        **log n**

b.         n

c. n2

d.         nn

**6.**               **The running time of an algorithm is given by T(n)=T(n1)+T(n-2)-T(n-3), if n>3 = n otherwise  
The order of this algorithm is**

**a.**        **n**

b.         log n

c. nn

d.         n2

**7.**               **If n=4,then the value of O(log n) is**

a.         1

**b.**        **2**

c. 4

d.         8

**8.**               **If n=4,then the value of O( n2) is**

a.         4**Rohith**

**b.**        **16**

c. 64

d.         512

**9.**               **The average time complexity of insertion sort is**

**a.**        **O(n2)**

b.         O(n)

c. O(1)

d.         O(log n)

**10.**        **The running time of an algorithm is given byT(n)=T(n-1)+T(n-2)-T(n-3), if n>3  
= n otherwise  
What should be the relation between T(1),T(2) and T(3) so that its order is constant.**

**a.**        **T(1)=T(2)=T(3)**

b.         T(1)+T(3)=2T(2)

c.T(1)-T(3)=T(2)

d.         T(1)+T(2)=T(3)

**11.**        **The order of the algorithm that finds a given Boolean function of `n' variables , produces a is**

a.         constant

b.         linear

c.non-linear

**d.**        **exponential**

**12.**        **If n=16, then the value of O(n log n) is**

a.         16

b.         32

**c.**         **64**

d.         128

**13.**        **How many memory management functions are there in C**

a.         4

b.         3

**c.**         **2**

d.         1

**14.**        **Which of the following is not a C memory allocation function**

**a.**        **alloc( )**

b.         calloc( )

c.free

d.         malloc()

**15.**        **If n= 8, then the value of O(1) is**

**a.**        **1**

b.         2

c. 4

d.         8

**16.**        **If n=4, then the value of O(n3) is**

a.         4

b.         16

**c.**         **64**

d.         512

**17.**        **If n=2, then the value of O(n) is**

**a.**        **2**

b.         3

c. 4

d.         5

**26.**        **Which of the following is considered auxiliary storage?**

**a.**        **disk**

b.         random access memeory(RAM)

c.read only memory(ROM)

d.         EEPROM

**30.**        **The linked list field(s) are**

a.         data

b.         pointer

c. pointer to next node

**d.**        **data and pointer to next node**

**31.**        **The linked list structure is defined as  Rohith**

**a.**        **struct node  
{  
int item;  
struct node \*next;  
};**

b.         node  
{  
int item;  
struct node \*next;  
};

c.struct node  
{  
int item;  
node \*node;  
};

d.         node  
{  
Int item;  
node next;  
};

**32.**        **Dynamic memory area is**

**a.**        **heap**

b.         stack

c.permanent storage

d.         Hard disk

**33.**        **The contents of the storage space allocated dynamically, can be accessed through \_ \_ \_ \_ \_ \_ \_**

a.         structure variables

**b.**        **pointers**

c. unions

d.         arrays

**34.**        **Each item in the list contains a �link� to structure containing the \_ \_ \_ \_ \_ \_ \_ item**

a.         previous

**b.**        **next**

c.present

d.         last

**35.**        **In C, program instructions are stored in**

a.         stack

b.         heap

**c.**         **permanent storage**

d.         Hard disk

**36.**        **In C, Global variables are stored in**

**a.**        **permanent storage**

b.         stack

c.heap

d.         Hard disk

**37.**        **In C, static variables are stored in**

a.         heap

**b.**        **permanent storage**

c.Hard disk

d.         Stack

**38.**        **A list refers to a set of items organized \_ \_ \_ \_ \_**

**a.**        **sequentially**

b.         exponentially

c.non-sequentially

d.         factorially

**39.**        **Each structure of a linked list consists \_ \_ \_ \_ \_ \_ \_ no. of fields**

**a.**        **2**

b.         3

c.4

d.         1

**40.**        **Linked lists are not suitable for data structures of which one of the following problem?**

a.         insertion sort

**b.**        **Binary search**

c.radix sort

d.         polynomial manipulation problem

**41.**        **An item that is read as input can be either pushed to a stack and latter popped and printed, or printed directly. Which of the following will be the output if the input is the sequence of items-1,2,3,4,5?**

a.         3,4,5,1,2

**b.**        **3,4,5,2,1**

c.1,5,2,3,4

d.         5,4,3,1,2

**42.**        **No.of pointers to be manipulated in a linked list to delete an item in the middle \_ \_ \_ \_ \_ \_ \_**

a.         Zero

b.         One

**c.**         **Two**

d.         Three

**43.**        **No.of pointers to be manipulated in a linked list to delete first item**

a.         Zero

**b.**        **One**

c.Two

d.         Three

**44.**        **Stack is useful for \_ \_ \_ \_ \_ \_ \_**

a.         radix sort

b.         breadth first search

**c.**         **recursion**

d.         quick sort

**45.**        **The end of the list is marked as**

**a.**        **node.next=0**

b.         (node.last = 0)

c.node.next= &node;

d.         node.previous=0;

**46.**        **No.of pointers to be manipulated in a linked list to insert an item in the middle \_ \_ \_ \_ \_ \_ \_ \_**

**a.**        **Two**

b.         Three

c.One

d.         Zero

**47.**        **No. of pointers to be manipulated in a linked list to delete last item**

a.         Zero

**b.**        **One**

c.Two

d.         Three

**48.**        **Single linked list uses \_ \_ \_ \_ \_ \_ no. of pointers**

a.         Zero

**b.**        **one**

c.Two

d.         Three

**49.**        **LIFO is**

**a.**        **stack**

b.         queue

c.linked list

d.         tree

**50.**        **A stack is has the entries a,b,c,(with a on top). Stack B is empty. An entry popped out of stack A can be printed immediately or pushed to stack B.An entry popped out of stack B can only be printed. In this arrangement, which of the following permutations a,b,c is not possible?**

a.         b a c

b.         b c a

**c.**         **c a b**

d.         a b c

**51.**        **Which of the following programming languages features require a stack-base allocation**

a.         pointer

b.         Block-structure

c.recursion

**d.**        **dynamic scoping**

**52.**        **Push down stack or push down list is**

**a.**        **stack**

b.         queue

c.linked list

d.         dequeue

**53.**        **Stack is useful for**

a.         radix sort

b.         breadth first search

**c.**         **recursion**

d.         Heap sort

**54.**        **Stacks can not be used to**

a.         evaluate an arithmetic expression in postfix form

b.         implement recursion

c.convert a given arithmetic expression in infix form to is equivalent postfix form

**d.**        **allocates resources (like CPU) by the operating system**

**55.**        **Stack is useful for implementing**

a.         radix sort

b.         breadth first search

c.selection sort

**d.**        **depth first search**

**56.**        **Which of the following is useful in implementing quick sort?**

**a.**        **stack**

b.         set

c.list

d.         queue

**57.**        **Which of the following is essential for converting an infix expression to postfix form efficiently?**

**a.**        **An operator stack**

b.         An operand stack

c.An operator stack and an operand stack

d.         A parse tree

**58.**        **A stack is most suitable to evaluate \_ \_ \_ \_ \_ expression**

**a.**        **postfix**

b.         prefix

c.infix

d.         post & infix

**59.**        **Linear linked data structure is**

a.         tree

b.         graph

**c.**         **stack**

d.         binary tree

**60.**        **A queue of characters currently contained a,b,c,d. What would be the contents of queue after the following operationDELETE, ADD W, ADD X, DELETE, ADD Y**

a.         A,B,C,W,Y

**b.**        **C,D,W,X,Y**

c.W,Y,X,C,D

d.         A,B,C,D,W

**61.**        **Which of the following data structure is suitable for priority queue?**

**a.**        **Doubly linked list**

b.         Circular queues

c.Binary search

d.         Heaps

**62.**        **For storing the sorted data on which often insert and deletion operations are performed, the following data structure is better**

a.         Array

b.         queue

c.linked-list

**d.**        **doubly linked-list**

**63.**        **A circular queue of size N will sign queue full when the number of elements in the queue is**

**a.**        **N-1**

b.         N

c.N+1

d.         N-2

**64.**        **The postfix equivalent of the prefix \* + a b - c d is**

**a.**        **ab + cd - \***

b.         ab cd + - \*

c.ab + cd \* -

d.         ab + - cd \*

**65.**        **The postfix expression for the infix expressionA + B\* (C+D) / F + D\*E is:**

a.         AB + CD + F / D + E\*

**b.**        **ABCD + \* F / + DE\***

c.A\*B + CD / F\*DE ++

d.         A+ BCD / F\* DE ++

**66.**        **A telephone system which places cells to a particular number on hold can best represented by**

**a.**        **queue**

b.         stack

c.linked-list

d.         variable

**67.**        **The performance of an algorithm is specified by the following notation that represents the worst case**

**a.**        **O-notation**

b.         Omega notation

c.Theta notation

d.         alpha-notation

**68.**        **If front=rear ,then the queue is**

a.         full

**b.**        **empty**

c.unknown value

d.         1/2 full

**69.**        **Reverse polish expression is**

a.         Infix

**b.**        **postfix**

c.prefix

d.         post & prefix

**70.**        **A list of integers is read in, one at a time, and a binary search tree is constructed. Next the tree is traversed and the integers are printed. Which traversed would result in a printout which duplicates the original order of the list of integers?**

a.         pre-order

b.         post-order

**c.**         **in-order**

d.         in-fix order

**71.**        **The postfix expression for the infix expression A + B\* (C+D) / F + D\*E is**

a.         AB + CD + \* F/D + E \*

**b.**        **ABCD + \*F / + DE\* +**

c.A\*B + CD / F\*DE ++

d.         A + \*BCD / F\*DE ++

**72.**        **The equivalent of (a+b↑c↑d)\*(e+f/d) in the post fix notation is**

a.         ab+c↑d↑e &fd/

b.         abcd↑+↑efd/+\*

c.abcdefd/+\*↑↑+

**d.**        **abcd↑↑+efd/+\***

**73.**        **The infix form of the postfix expression ABC-/D\*E+ is**

a.         A/B-C\*D+E

b.         A-B/C\*D+E

c.(A-B)/C\*D+E

**d.**        **A/(B-C)\*D+E**

**74.**        **The postfix expression for the infix expression A/B\*C+D\*E is**

**a.**        **AB/C\*DE\*+**

b.         ABC/\*DE+\*

c.ABCD/\*E+\*

d.         ABC\*/D\*E+

**75.**        **The prefix expression for the infix expressionA/B\*C+D\*E is**

a.         AB/C\*DE\*+

**b.**        **+\*/ABC\*DE**

c.+\*AB/C\*DE

d.         /+ABCDE

**76.**        **Suffix expression is**

a.         Infix

**b.**        **postfix**

c.prefix

d.         post & prefix

**77.**        **polish expression is**

**a.**        **infix**

b.         postfix

c.prefix

d.         post & prefix

**78.**        **To convert an Infix expression into postfix we require**

**a.**        **stack**

b.         queue

c.linked list

d.         dequeue

**79.**        **A stack is most suitable to evaluate \_ \_ \_ \_ \_ \_ \_ expression**

**a.**        **postfix**

b.         prefix

c.infix

d.         post & infix

**80.**        **The circular linked list have**

a.         no beginning

b.         no ending

c.beginning but no ending

**d.**        **no beginning and no ending**

**81.**        **To insert a node at the beginning of the doubly linked list \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

**b.**        **2**

c.3

d.         4

**82.**        **Doubly linked list uses \_ \_ \_ \_ \_ \_ \_ \_ no.of pointers**

a.         Zero

b.         One

**c.**         **Two**

d.         Three

**83.**        **To insert a node at the beginning of the single linked list \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

**a.**        **1**

b.         2

c.3

d.         0

**84.**        **To insert a node at middle of the single linked list \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

**a.**        **1**

b.         2

c.3

d.         4

**85.**        **To insert a node at the end of the doubly linked list \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

**b.**        **2**

c.3

d.         4

**86.**        **To insert a node at the end of the single linked list \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

**a.**        **1**

b.         2

c.3

d.         4

**87.**        **To delete the first node in single linked list \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

**a.**        **1**

b.         2

c.3

d.         4

**88.**        **To delete the last node in single linked list \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

**a.**        **1**

b.         2

c.3

d.         0

**89.**        **To delete the middle node in single linked list \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

**b.**        **2**

c.3

d.         4

**90.**        **To delete an item in the middle of a circular doubly linked list, \_ \_ \_ \_ \_ \_ \_ \_ no.of points to be manipulated**

a.         2

b.         4

**c.**         **6**

d.         8

**91.**        **If storage class is missing in the array definition, by default it will be taken to be**

a.         automatic

b.         external

c.static

**d.**        **either automatic or external depending on the place of occurrence**

**92.**        **To delete the last node in doubly linked list \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

**b.**        **2**

c.3

d.         4

**93.**        **To delete the middle node in doubly linked list \_ \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

b.         2

c.3

**d.**        **4**

**94.**        **To insert an item in a circular doubly linked list, \_ \_ \_ \_ \_ \_ \_ no.of points to be manipulated**

a.         1

b.         2

c.3

**d.**        **4**

**95.**        **Which of the following features of C is meant to provide reliable access to special memory**

**a.**        **static \_ const**

b.         pragma

c.volatile

d.         immutable

**96.**        **To insert a node at middle of the doubly linked list \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

b.         2

c.3

**d.**        **4**

**97.**        **To delete the first node in doubly linked list \_ \_ \_ \_ \_ \_ \_ \_ no. of pointers to be manipulated**

a.         1

**b.**        **2**

c.3

d.         4

**98.**        **To insert an item in a circular single linked list \_ \_ \_ \_ \_ \_ \_ \_ \_ no.of points to be manipulated**

**a.**        **2**

b.         3

c.4

d.         1

**99.**        **To delete an item in a circular doubly linked list, \_ \_ \_ \_ \_ \_ \_ \_ no.of points to be manipulated**

a.         1

b.         2

c.3

**d.**        **4**

**100.**  **A sorting technique is called stable if:**

a.         it takes O ( n log n) time

**b.**        **It maintains the relative order of occurrence of non-distinct elements**

c.It uses divide and conquer paradigm

d.         The maximum number of nodes in a binary tree of height h is (2-1)(The height of the root is reckoned as 0)

**101.**  **The maximum number of comparisons needed to sort 7 items using radix sort is (assume each item is a 4 digit decimal number)**

**a.**        **280**

b.         40

c.47

d.         38

**102.**  **If each node in a tree has a value greater than every value in its left sub tree and has value less than every value in its right sub tree, the binary tree is known as**

a.         Complete binary tree

b.         Full binary tree

**c.**         **Binary search tree**

d.         Threaded binary tree

**103.**  **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 as possible, is known as**

a.         full binary tree

b.         2-tree

c.threaded tree

**d.**        **Complete binary tree**

**104.**  **You are asked 15 randomly generated numbers. You should prefer**

**a.**        **bubble sort**

b.         quick sort

c.merge sort

d.         heap sort

**105.**  **Which data structure is needed to convert infix notation to post fix notation**

a.         B-tee

b.         Queue

c.Tree

**d.**        **Stack**

**106.**  **The time required to search an element in a binary search tree having n elements is**

a.         O(1)

**b.**        **O(log2 n)**

c.O(n)

d.         O(n log2 n)

**107.**  **A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is**

a.         log2 n

**b.**        **n-1**

c.n

d.         2n

**108.**  **A tree, for which at every node the height of its left sub tree and right sub tree differ at most by one is a/an**

a.         Binary search tree

**b.**        **AVL tree**

c.Complete binary tree

d.         Threaded binary tree

**109.**  **Which of the following sorting algorithms does not have a worst case running time complexity of O(n2)?**

a.         Insertion sort

**b.**        **Merge sort**

c.Quick sort

d.         Bubble sort

**110.**  **Which of the following is not a correct statement**

a.         internal sorting is used if the number of items to be sorted is very large

b.         External sorting is used if the number of items to be sorted is very large

c.External sorting needs auxiliary storage

**d.**        **Internal sorting needs auxiliary storage**

**111.**  **There are 4 different algorithms A1,A2,A3,A4 to solve a given problem with the order log(n),log(log(n)),nlog(n),n/log(n) respectively. Which is the best algorithm?**

a.         A1

**b.**        **A2**

c.A3

d.         A4

**112.**  **Which of the following algorithms exhibits the unusual behavior that, minimum numbers of comparisons are needed if the list to be sorted is in the reverse order and maximum numbers of comparisons are needed if they are already in sorted order?**

a.         Heap tree

b.         Radix sort

**c.**         **Binary insertion sort**

d.         Selection sort

**113.**  **You want to check whether a given set of items is sorted. Which of the following sorting methods will be the most efficient if it is already in sorted order?**

a.         bubble sort

b.         selection sort

**c.**         **insertion sort**

d.         merge sort

**114.**  **The way a card game player arranges his cards as he picks them up one by one , is an example of**

a.         bubble sort

b.         selection sort

**c.**         **insertion sort**

d.         merge sort

**115.**  **Which of the following sorting algorithm has the worst time complexity of nlog(n)?**

**a.**        **Heap sort**

b.         Quick sort

c.Insertion sort

d.         Selection sort

**116.**  **Which of the following sorting methods sorts a given set of items that is already in sorted order or in reverse sorted order with equal speed?**

a.         Heap sort

**b.**        **Quick sort**

c.Insertion sort

d.         Selection sort

**117.**  **Which of the following sorting methods will be the best if number of swapping done, is the only measure of efficiency?**

a.         bubble sort

**b.**        **insertion sort**

c.selection sort

d.         heap sort

**118.**  **As part of the maintenance work, you are entrusted with the work of rearranging the library books in a shelf in proper order, at the end of each day. The ideal choice will be**

a.         bubble sort

**b.**        **insertion sort**

c.selection sort

d.         heap sort

**119.**  **Sorting is not useful for**

a.         report generation

**b.**        **minimizing the storage needed**

c.making searching easier and efficient

d.         responding to queries easily

**120.**  **A machine took 200 sec to sort 200 names, using bubble sort. In 800 sec. it can approximately sort \_ \_ \_ \_ \_ \_ \_ \_ \_ names**

**a.**        **400**

b.         800

c.750

d.         1600

**121.**  **A machine needs a minimum of 100 sec. to sort 1000 names by quick sort. The minimum time needed to sort 100 names will be approximately**

a.         50.2 sec

**b.**        **6.7 sec**

c.72.7 sec.

d.         11.2 sec.

**122.**  **A sorting method with \_ \_ \_ \_ \_ \_ \_ \_ is the most efficient method**

**a.**        **O(log n)**

b.         O(n)

c.O(1)

d.         O(n2)

**123.**  **Which of the following statement is false?**

a.         Optimal binary search construction can be performed efficiently using dynamic programming

b.         Breadth-first search cannot be used to find connected components of a graph

**c.**         **Given the prefix and postfix walks of a binary tree, the binary cannot be uniquely reconstructed**

d.         Depth-first search can be used to find the connected components of a graph

**124.**  **The average successful search time for sequential search on 'n' items is**

a.         n/2

b.         (n-1)/2

**c.**         **(n+1)/2**

d.         log(n)+1

**125.**  **A hash function f defined as f(key)=key mod 7, with linear probing, is used to insert the keys 37,38,72,48,98,1,56, into a table indexed from 0 to 6. What will be the location of key 11?**

a.         3

b.         4

**c.**         **5**

d.         6

**126.**  **The order of the binary search algorithm is**

**a.**        **n**

b.         n2

c.nlog(n)

d.         log(n)

**127.**  **Linked lists are not suitable for implementing**

a.         insertion sort

**b.**        **binary search**

c.radix sort

d.         polynomial manipulation

**128.**  **Stack is useful for**

a.         radix sort

b.         breadth first search

c.heap sort

**d.**        **depth first search**

**129.**  **Which of the following algorithm design technique is used in the quick sort algorithm?**

a.         Dynamic programming

b.         Backtracking

**c.**         **Divide and conquer**

d.         Greedy method

**130.**  **The average successful search time taken by binary search on a sorted order array of 10 items is**

a.         2.6

b.         2.7

c.2.8

**d.**        **2.9**

**131.**  **A 3-ary 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**

a.         10

b.         17

c.23

**d.**        **13**

**132.**  **Which of the following traversal techniques lists the nodes of a binary search tree in ascending order?**

a.         post-order

**b.**        **In-order**

c.Pre-order

d.         No-order

**133.**  **A general linear list is a list in which operations, such as retrievals, insertions, changes, and deletions can be done \_ \_ \_ \_ \_ \_ \_ \_ \_**

**a.**        **any where in the list**

b.         only at the beginning

c.only at the end

d.         only at the middle

**134.**  **A(n) \_ \_ \_ \_ \_ \_ \_ is a collection of elements and relationship Among them.**

a.         abstract data type

b.         array

**c.**         **data structure**

d.         standard type

**135.**  **Data that consists of a single, non decomposable entity are known as \_ \_ \_ \_ \_ \_**

**a.**        **atomic data**

b.         array

c.data structure

d.         standard type

**136.**  **A binary tree has n leaf nodes. The number of nodes of degree 2 in this tree is**

a.         logn

**b.**        **n-1**

c.n

d.         2n

**137.**  **A full binary tree with n leaf nodes contains**

a.         n nodes

b.         log2 n nodes

**c.**         **2n-1 nodes**

d.         2n nodes

**138.**  **The number of binary trees with 3 nodes which when traversed in post-order gives the sequence A,B,C is**

a.         3

b.         9

c.7

**d.**        **5**

**139.**  **Which of the following need not be a binary tree?**

a.         Search tree

b.         Heap

c.AVL-tree

**d.**        **B-tree**

**140.**  **A binary tree in which every non-leaf node has non-empty left and right subtrees is called a strictly binary tree.Such a tree with 10 leaves**

a.         cannot be more than 19 nodes

**b.**        **has exactly 19 nodes**

c.has exactly 17 nodes

d.         can not have more than 17 nodes

**141.**  **Find the odd man out**

a.         binary tree

b.         Avl tree

c.graph

**d.**        **queue**

**142.**  **The depth of a complete binary tree with n nodes(log is to the base two)**

**a.**        **log(n+1)-1**

b.         log(n)

c.log(n+1)+1

d.         log(n)+1

**143.**  **The following is an example of a non-linear data structure**

a.         stack

b.         queue

**c.**         **tree**

d.         linear list

**144.**  **If a graph is represented as a linked list, \_ \_ \_ \_ \_ \_ \_ \_ \_ no.of list nodes are required**

**a.**        **1**

b.         2

c.3

d.         4

**145.**  **The number of possible binary trees with 4 nodes is**

a.         12

**b.**        **14**

c.13

d.         15

**146.**  **The number of possible binary trees with 3 nodes is**

a.         12

b.         13

**c.**         **5**

d.         15

**147.**  **The number of possible ordered trees with 3 nodes A,B,C is**

a.         16

**b.**        **12**

c.6

d.         10

**148.**  **A tree is a \_ \_ \_ \_ \_ data structure**

a.         non-recursive

**b.**        **recursive**

c.linear

d.         non-linear

**149.**  **A node that does not have any sub-tree is called a \_ \_ \_ \_ \_ \_ \_**

**a.**        **terminal node**

b.         root node

c.left node

d.         right node

**150.**  **The number of edges in a regular graph of degree d and n vertices is**

a.         maximum of n, d

b.         n+d

c.nd

**d.**        **nd/2**

**151.**  **Which of the following algorithms solves the all pair shortest path problem?**

a.         Diskstra's algorithm

**b.**        **Floyd algorithm**

c.Prim's algorithm

d.         Warshall's algorithm

**152.**  **The minimum number of colors required to color a graph having n (n>3) vertices and 2 edges is**

a.         4

b.         3

**c.**         **2**

d.         1

**153.**  **The maximum degree of any vertex in a simple graph with n vertices is**

a.         n

**b.**        **n-1**

c.n+1

d.         2n-1

**154.**  **A graph G with n nodes is bipartite if it contains**

**a.**        **n edges**

b.         a cycle of odd length

c.no cycle of odd length

d.         n2 edges

**155.**  **A graph can be represented as an \_ \_ \_ \_ \_ \_**

**a.**        **Linked list**

b.         Structure

c.Union

d.         Queue

**156.**  **A graph can be represented as an \_ \_ \_ \_ \_ \_**

**a.**        **Array**

b.         Structure

c.Union

d.         Queue

**157.**  **The minimum number of edges in a connected cyclic on n vertices is**

a.         n-1

**b.**        **n**

c.n+1

d.         n+2

**158.**  **Which of he following is useful in traversing a given graph by breadth first search?**

a.         Stack

b.         Set

c.List

**d.**        **Queue**

**159.**  **Sparse matrices have**

**a.**        **many zero entries**

b.         many non-zero entries

c.higher dimensions

d.         lower dimensions

**160.**  **The maximum no.of edges in an undirected graph with out loops with n vertices is**

a.         n

b.         n\*(n-1)

**c.**         **n\*(n-1)/2**

d.         n-1

**161.**  **Which of the following abstract data types can be used to represent a many to many relationship**

a.         tree

**b.**        **graph**

c.queue

d.         stack

**162.**  **In a directed graph without self loops with n verices , the maximum no.of edges is**

a.         n

**b.**        **n\*(n-1)**

c.n\*(n-1)/2

d.         n-1

**163.**  **An n vertex undirected graph with exactly n\*(n-1)/2 edges is said to be**

**a.**        **Complete graph**

b.         Un complete graph

c.Directed graph

d.         Un directed graph

**164.**  **To create a node dynamically in a singly linked list \_ \_ function in C is used**

**a.**        **malloc()**

b.         calloc()

c.alloc()

d.         dealloc()

**165.**  **In an undirected graph, the sum of degrees of all the nodes**

**a.**        **must be even**

b.         is thrice the number of edges

c.must be odd

d.         need not be even

**166.**  **In an undirected graph, the sum of degrees of all the nodes**

a.         is thrice the number of edges

**b.**        **is twice the number of edges**

c.must be odd

d.         need not be even

**167.**  **\_ \_ \_ function is used to in C to dynamically allocate space for more than one object**

a.         malloc()

**b.**        **calloc()**

c.alloc()

d.         dealloc()

**168.**  **\_ \_ \_ function is used to in C to dynamically allocate space for one object**

**a.**        **malloc()**

b.         calloc()

c.alloc()

d.         dealloc()

**169.**  **If n=2, then the value of O(n log n) is**

**a.**        **2**

b.         4

c.8

d.         16

**170.**  **Calloc(m,n); is equivalent to**

a.         malloc(m\*n,0);

b.         memset(0,m\*n);

**c.**         **ptr=malloc(m\*n);memset(p,0,m\*n)**

d.         ptr=malloc(m\*n);strcpy(p,0)

**171.**  **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 are**

**a.**        **2,2,1,1,2**

b.         2,2,1,2,2

c.2,1,2,2,1

d.         2,1,2,2,2

**172.**  **return type of a realloc( ) function is**

a.         int

b.         float

c.char

**d.**        **void**

**173.**  **To delete an element from a queue we use the \_ \_ \_ \_ \_ operation**

a.         pop

b.         push

c.enqueue

**d.**        **dequeue**

**174.**  **To add an element to a queue we use the \_ \_ \_ \_ \_ operation**

a.         pop

b.         push

**c.**         **enqueue**

d.         dequeue

**175.**  **Which of the memory function allocates a contiguous memory**

a.         malloc( )

**b.**        **calloc( )**

c.release( )

d.         free( )

**176.**  **Return type of a malloc( ) function is**

a.         int

b.         float

c.char

**d.**        **void**

**177.**  **A queue is a \_ \_ \_ \_ \_ \_ structure**

a.         first in-last out

b.         lasting-first-out

**c.**         **first in-first out**

d.         last in-last out

**178.**  **A queue is a list in which insertion can be done \_ \_ \_ \_**

a.         any where in the list

b.         only at the beginning

**c.**         **only at the end**

d.         only at the middle

**179.**  **A \_ \_ \_ \_ \_ \_ is a first in - last out(FIFO) data structure in which insertions are restricted to one end, called the rear, and deletions are restricted to another end ,called the front**

a.         Stack

**b.**        **queue**

c.tree

d.         binary tree

**180.**  **The pointer(s) in a queue points to**

a.         start of the queue

b.         end of the queue

c.middle of the queue

**d.**        **both start and end of the queue**

**181.**  **The disadvantage of the queue is**

**a.**        **when the item is deleted, the space for that item is not claimed**

b.         when the item is deleted, the space for that item is claimed

c.a non destructive

d.         increases the memory space

**182.**  **A queue is a list in which deletion can be done \_ \_ \_ \_**

a.         any where in the list

**b.**        **only at the beginning**

c.only at the end

d.         only at the middle

**183.**  **Read() operation in queue is**

a.         non-destructive

b.         additive

c.push()

**d.**        **destructive**

**184.**  **In which of the data structure, space for the item is not claimed ,when an item is deleted**

**a.**        **queue**

b.         circular queue

c.stack

d.         linked list

**185.**  **As the items from a queue get deleted, the space for item is not reclaimed in queue. This problem is solved by**

**a.**        **circular queue**

b.         stack

c.linked list

d.         doubly linked list

**186.**  **Which of the following operation is used to add an item in a queue**

**a.**        **write()**

b.         read()

c.pop()

d.         push()

**187.**  **\_ \_ \_ \_ no.of pointers are required to implement read and write operations in a queue**

**a.**        **two**

b.         three

c.four

d.         five

**188.**  **FIFO is**

a.         stack

**b.**        **queue**

c.linked list

d.         tree

**189.**  **Which of the following operation is used to an item in a queue**

a.         write()

**b.**        **read()**

c.pop()

d.         push()

**190.**  **The number of swapping needed to sort the numbers 8,22,7,9,31,19,5,13 in an ascending order, using bubble sort is**

a.         11

b.         12

c.13

**d.**        **14**

**191.**  **Given two sorted list of size 'm' and 'n' respectively. The number of comparisons needed by the merge sort algorithm will be**

a.         m x n

b.         maximum of m,n

**c.**         **minimum of m,n**

d.         m+n-1

**192.**  **For merging two sorted lists of sizes m and n into a sorted list of size m+n, requires \_ \_ \_ \_ \_ \_ \_ \_ no.of comparisons**

a.         O(m)

b.         O(n)

**c.**         **O(m+n)**

d.         O(log(m)+log(n))

**193.**  **The principle of locality justifies the use of**

a.         interrupts

b.         DMA

c.polling

**d.**        **cache memory**

**194.**  **The concatenation of two lists is to be performed in O(1) time. Which of the following implementations of a list could be used?**

a.         Singly linked list

b.         Doubly linked list

**c.**         **Circularly doubly linked list**

d.         Array implementation of list

**195.**  **The initial condition of a queue is**

**a.**        **front=rear=-1**

b.         front=rear

c.  front=rear=n

d.         front=rear=1

**196.**  **A sorting technique that guarantees , that records with the same primary key occurs in the same order in the sorted list as in the original unsorted list is said to be**

**a.**        **stable**

b.         consistent

c.  external

d.         linear

**197.**  **The average number of comparisons performed by the merge sort algorithm , in merging two sorted lists of length 2 is**

**a.**        **8/3**

b.         8/5

c.  11/7

d.         1/16

**198.**  **Merge sort uses**

**a.**        **divide and conquer strategy**

b.         backtracking approach

c.  heuristic approach

d.         greedy approach

**199.**  **Queue can be used to implement**

**a.**        **radix sort**

b.         quick sort

c.  recursion

d.         depth first search