UNIT-X

DATA STRUCTURE AND PROGRAMMING

1. Data Structure and Programming 1-30

NOTES

CHAPTER 1

DATA STRUCTURE AND PROGRAMMING

OBJECTIVE QUESTIONS

1. The time complexity of linear search algorithm over an array of n elements is.
   (a) O ($\log_2 n$) (b) O (n)
   (c) O ($n \log_2 n$) (d) O (n^2)
2. The average time required to perform a successful sequential search for an element in an array A (1 : n) is given by.
   (a) $\frac{n+1}{2}$ (b) $\frac{n(n+1)}{2}$
   (c) $\log_2 n$ (d) n^2
3. Which of the following is false?
   (a) A serial search begins with the first array element
   (b) A serial search continues searching, element by element, either until a match is found or until the end of the array is encountered.
   (c) A serial search is useful when the amount of data that must be search is small.
   (d) For a serial search to work, the data in the array must be arranged in either alphabetical or numerical order.
4. To sort many large object or structures, it would be most efficient to
   (a) place reference to them in and array an sort the array
   (b) place them in a linked list and sort the linked list
   (c) place pointers to them in an array and sort the array
   (d) place them in an array and sort the array
5. Minimum number of comparison required to compute the largest and second largest element in an array is.
   (a) $n - [\log_2 n] - 2$ (b) $n + [\log_2 n - 2]$
   (c) $\log_2 n$ (d) none of these
6. Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main()
{
static int arr[] = {0, 1, 2, 3, 4};
int *p[] = {arr, arr + 1, arr + 2, arr + 3, arr + 4};
int **ptr = p;
ptr++ ;
printf ("%d %d %d\n", ptr - p, *ptr - arr, **ptr);
*ptr++ ;
printr ("%d %d %d\n", ptr - p, *ptr - arr, ** ptr);
*++ptr;
printr ("%d %d %d\n", ptr - p, *ptr - arr, ** ptr);
++*ptr;
printr ("%d %d %d\n", ptr - p, *ptr - arr, ** ptr);
return 0;
}
```

```
(a) 0 0 0        (b) 1 1 2
    1 1 1            2 2 3
    2 2 2            3 3 4
    3 3 3            4 4 1
(c) 1 1 1        (d) 0 1 2
    2 2 2            1 2 3
    3 3 3            2 3 4
    3 4 4            3 4 Garbage
```

7. What will be the output of the following program, if the array begins at address 1898320?

```
#include <stdio.h>
int main()
{
    int arr[] = {12, 14, 15, 23, 45};
    printf ("%u %u %u %u\n", arr, &arr, arr +
1,&arr
    + 1);
    return 0;
}
```

8. What will be the output of the following program?

```
#include <stdio.h>
int main()
{
    float a[] = {12.4, 2.3, 4.5, 6.7};
    printf ("%d\n", sizeof (a)/sizeof (a[0]));
    return 0;
}
```

9. Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main()
{
    int a[2][2] = {1, 2, 3, 4};
    int i, j;
    int *p[] = { (int *) a, (int *) a + 1, (int *) a
+ 2};
    for (i = 0; i < 2; j++)
    {
    for (j = 0; j < 2; j++)
    printf ("%d %d %d %d\n",
    *(* (p + i) + j),
    *(* (j + p) + i),
    *(* (i + p) + j),
    *(*(p + j) + i));
    }
    return 0;
}
```

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

10. Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main()
{
    void fun (int, int []);
    int arr[] = {1, 2, 3, 4};
    int i;
    fun (4, arr);
    for (i = 0; i < 4 ; i++)
                    printf ("%d", arr[i]);
    printf ("\n");
    return 0;
}
void fun (int n, int arr[])
{
    int *p;
    int i;
    for (i = 0; i < n ; i++)
                    p = &arr [i];
    *p = 0;
}
```

(a) 2 3 4 5 (b) 1 2 3 0
(c) 0 1 2 3 (d) 3 2 1 0

11. Which of these best describes an array?

(a) A data structure that shows a hierarchical behavior
(b) Container of objects of similar types
(c) Container of objects of mixed types
(d) All of the mentioned

12. How do you initialize an array in C?

(a) intarr[3] = (1,2,3);
(b) intarr(3) = {1,2,3};
(c) intarr[3] = {1,2,3};
(d) intarr(3) = (1,2,3);

13. What are the advantages of arrays?

(a) Easier to store elements of same data type

(b) Used to implement other data structures like stack and queue

(c) Convenient way to represent matrices as a 2D array

(d) All of the mentioned

14. What are the disadvantages of arrays?

(a) We must know before hand how many elements will be there in the array

(b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size

(c) Insertion and deletion becomes tedious

(d) All of the mentioned

15. Assuming int is of 4bytes, what is the size of intarr[15];?

(a) 15 (b) 19

(c) 11 (d) 60

16. What is the time complexity for inserting/deleting at the beginning of the array?

(a) O(1) (b) O(n)

(c) O(logn) (d) O(nlogn)

17 The expression 5 – 2 – 3 * 5 – 2 will evaluate to 18, if – is left associative and

(a) * has precedence over *

(b) * has precedence over –

(c) – has precedence over *

(d) – has precedence over –

18. Which of the following is a collection of items into which items can be inserted arbitrarity and from which only the smallest item can be removed?

(a) Descending priority queue

(b) Ascending priority queue

(c) Fifo queue

(d) Lifo queue

19. The five items: A, B, C, D, and E are pushed in a stack, one after the other starting from A. The stack is popped four times and each element is inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack. Now one item is popped from the stack. The popped item is.

(a) A

(b) B

(c) C

(d) D

20. Using Pop (SI, Item), Push (SI, Item), Read (Item), Print (Item), the variables SI (stack) and Item, and given the input file:

A, B, C, D, E, F < EOF>

which stacks are possible:

(a) 5	A		(b) 5	
4	B		4	
3	C		3	D
2	D		2	A
1	E		1	F
(c) 5			(d) 5	
4			4	
3	F		3	C
2	D		2	E
1	B		1	B

21. A posfix expression is merely the reverse of the prefix expression.

(a) True

(b) False

22. What can be said about the array representation of a circular queue when it contains only one element?

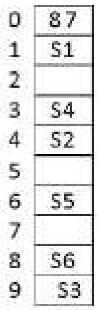
(a) front = Rear = Null

(b) front = Rear + 1

(c) front = Rear – 1

(d) None of these

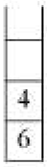
23. Following sequence of operations is performed on a stack push (1), push (2), pop, push (1), push (2), pop, pop, pop, push (2), Pop. The sequence of popped out values are

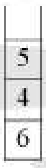


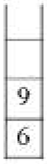
(a) 2, 2, 1, 1, 2 (b) 2, 2, 1, 2, 2

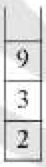
(c) 2, 1, 2, 2, 1 (d) 2, 1, 2, 2, 2

24. In evaluating the arithmetic expression 2*3 – (4 + 5), using stacks to evaluate its equivalent postfix form, which of the following stack configuration is not possible?

(a) 

(b) 

(c) 

(d) 

25. Stack A 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 the stack B can only be printed. In this arrangement, which of the following permutations of a, b, c are not possible?

(a) b a c (b) b c a

(c) c a b (d) a b c

26. Stacks can't be used to

(a) Evaluate an arithmetic expression in postfix form

(b) Implement recursion

(c) Convert a given arithmetic expression in infix form to its equivalent postfix form

(d) Allocate resources (like CPU) by the operating system.

27. Which one the following permutations can be obtained in the output (in the same order), using a stack assuming that the input is the sequence 1, 2, 3, 4, 5 in that order?

(a) 3, 4, 5, 1, 2 (b) 3, 4, 5, 2, 1

(c) 1, 5, 2, 3, 4 (d) 5, 4, 3, 2, 1

28. Consider a linked list implementation of a queue with two pointers: front and rear. The time needed to insert element in a queue of length n is

(a) O(1) (b) $O(\log_2 n)$

(c) O(n) (d) $O(n \log_2 n)$

29. A linear list in which elements can be added or removed at either end but not in the middel is called.

(a) queue (b) deque

(c) stack (d) tree

30. Stack is useful for implementing

(a) radix (b) breadth first search

(c) recursion (d) none of these

31. To insert a node in a circular list at rear position, it should be inserted at ____of the queue.

(a) Front position (b) Front – 1 position

(c) Rear position (d) Rear – 1 position

32. Let "q" be the queue of integer defined as follows:

```
# define MAX-Q 500
    struct queue {
    int item [MAX-Q];
    int front, rear;
    } q;
```

To insert an element in the queue, we may write operation

(a) ++q. item [q. rear] = X;

(b) q. item [q. rear] ++ = X;

(c) q. item [++ q. rear] = X;

(d) None of these

33. Which of the following data structure may give overflow error, even though the current number of element in it is less than its size?

(a) Simple queue (b) Circular queue

(c) Stack (d) None of these

34. Number of "ADD" and "REMOVE" operations required to access n/2th element of a queue of "n" elements so that the original queue remain the same after the access is (take help of another queue)

(a) 4 * n (b) 8 * n

(c) 4 * n – 1 (d) 8 * n – 1

35. The following postfix expression with single digit operands is evaluated using a stack

8 2 3 ^ / 2 3 * + 5 1 * –

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are.

(a) 6, 1 (b) 5, 7

(c) 3, 2 (d) 1, 5

36. 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

37. 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

38. Which of the following is an infix expression?

(a) (a+b)*(c+d)

(b) ab+c*

(c) +ab

(d) abc+*

39. What is the time complexity of an infix to postfix conversion algorithm?

(a) O(N log N)

(b) O(N)

(c) O(N2)

(d) O(M log N)

40. 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)*

41. Parentheses are simply ignored in the conversion of infix to postfix expression.

(a) True (b) False

42. It is easier for a computer to process a postfix expression than an infix expression.

(a) True (b) False

43 What is the postfix expression for the infix expression?

a-b-c

(a) -ab-c (b) abc-

(c) -abc (d) -ab-c

44. 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^/-

45. 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

46. 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

47. 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/

48. 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) abc*def^/g*-h*+

(c) abcd*^ed/g*-h*+

(d) abc*de^fg/*-*h+

49. What is the correct postfix expression for the following expression?

a+b*(c^d-e)^(f+g*h)-i

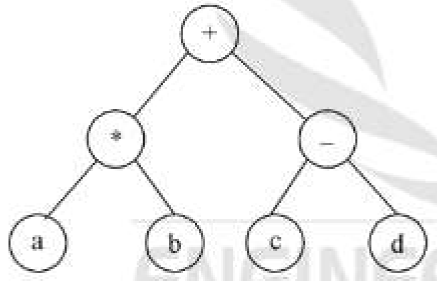
(a) abc^de-fg+*^*+i-

(b) abcde^-fg*+*^h*+i-

(c) abcd^e-fgh*+^*+i-

(d) ab^-dc*+ef^gh*+i-

50. From the given Expression tree, identify the correct postfix expression from the list of options.



(a) ab*cd*+ (b) ab*cd-+

(c) abcd-*+ (d) ab*+cd-

51. 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

52. 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)

53. The time required to search an element in a linked list of length n is.

(a) O ($\log_2 n$) (b) O (n)

(c) O (n) (d) O (n^2)

54. The worst case time required to search a given element in a sorted linked list of length n is.

(a) O (1) (b) O ($\log_2 n$)

(c) O (n) (d) O ($n \log_2 n$)

55. Consider a linked list of n element which is pointed by an external pointer. What is the time taken to delete the element which is successor of the element pointed to by a given pointer?

(a) O (1) (b) O ($\log_2 n$)

(c) O (n) (d) O ($n \log_2 n$)

56. Consider a linked list of n elements. What is the time taken to insert an element an after element pointed by some pointer?

(a) O (1) (b) O ($\log_2 n$)

(c) O (n) (d) O ($n \log_2 n$)

57. A linear collection of data element the linear node is given by mean of pointer is called.

(a) linked list

(b) node list

(c) primitive list

(d) none of these

58. The process of accessing data stored in a tape is similar to manipulating data on a

(a) stack (b) queue

(c) list (d) heap

59. When a new element is inserted in the middle of a linked list, then

(a) only elements that appear after the new element need to be moved

(b) only elements that appear before the new element need to be moved.

(c) elements that appear before and after the new element need to be moved.

(d) none of these

60. If address of the 8th element in a linked list of integers is 1022, then address of 9th element is.

(a) 1024 (b) 1026

(c) 1023 (d) none of these

61. Which of the following operations is performed more efficiently by doubly linked list than by linear linked list?

(a) Deleting a node whose location is given

(b) Searching an unsorted list for a given item

(c) Inserting a node after the node with a given location

(d) Traversing the list to process each node.

62. Consider a linked list implementation of a queue with two pointers: front and rear. The time needed to insert element in a queue of length n is

(a) O(1) (b) $O(\log_2 n)$

(c) O(n) (d) $O(n \log_2 n)$

63. A linear list in which elements can be added or removed at either end but not in the middle is called.

(a) queue (b) deque

(c) stack (d) tree

64. Queues serve a major role in

(a) simulation of recursion

(b) simulation of arbitrary linked list

(c) simulation of limited resource allocation

(d) expression evaluation

65. In a circularly linked list organization, insertion of a record involves the modification of

(a) no pointer

(b) 1 pointer

(c) 2 pointers

(d) 3 pointers

66. Given two statements:

(i) Insertion of an element should be done at the last node in a circular list.

(ii) Deletion of an element should be done at the last node of the circular list.

(a) Both are True

(b) Both are False

(c) First is true and second is false

(d) First is false and second is true

67. To free which of the following list traversing through the entire list is not necessary?

(a) Circular list

(b) Singly linked list

(c) Double linked list

(d) Both (b) and (c)

68. which of the following statement(s) is/are true regrading insertion of node in a linear linked list?

(a) Setting the field of the new node means allocating memory to newly created node

(b) If node precedes all others in the list, then insert it at the front and return its address

(c) Creating a new node depends upon free memory space

(d) All of these

69. Which of the following statements are true about a doubly linked list?

(a) It may be either linear or circular

(b) It must contain a header node

(c) It will occupy same memory space as that of linear linked list, both having same number of nodes

(d) All of these

70. Identity the steps to be taken when a first node is to be deleted from linear linked list.

I. Set link of start pointer to the second node in the list.

II. Free the space associated with first node

III. Obtain the address of the second node in the list.

IV. Count the number of nodes in the list.

(a) I and II (b) I, II and III

(c) II and III (d) I, II, III and IV

71. The concatenation of two lists is to be performed in O(1) time. Which of the following implementations of list should be used?

(a) Singly linked list

(b) Doubly linked list

(c) Circular doubly linked list

(d) Array implementation of list

72. What is the output of following function for start pointing to first node of following linked list?

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$

```
void fun(struct node* start)
{
if(start ==NULL)
return;
printf("%d ", start→data);
if(start→next !=NULL)
fun(start→next→next);
printf("%d ", start→data);
}
```

(a) 1 4 6 6 4 1

(b) 1 3 5 1 3 5

(c) 1 2 3 5

(d) 1 3 5 5 3 1

73. The following C function takes a simply-linked list as input argument.

It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank. Choose the correct alternative to replace the blank line.

```
typedef struct node
{
int value;
struct node *next;
}Node;
Node *move_to_front(Node *head)
{
Node *p, *q;
if((head ==NULL:||(head→next ==NULL))
return head;
q =NULL; p = head;
while(p→ next !=NULL)
{
q = p;
p = p→next;
}
______________________
return head;
}
```

(a) q = NULL; p→next = head; head = p;

(b) q→next = NULL; head = p; p→next = head;

(c) head = p; p→next = q; q→next = NULL;

(d) q→next = NULL; p→next = head; head = p;

74. The following C function takes a single-linked list of integers as a parameter and rearranges the elements of the list.

The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node
{
int value;
struct node *next;
};
void rearrange(struct node *list)
{
struct node *p, * q;
int temp;
if((!list)||!list→next)
return;
p = list;
q = list→next;
while(q)
{
temp = p→value;
p→value = q→value;
q→value = temp;
p = q→next;
q = p?p->next:0;
}
}
```

(a) 1, 2, 3, 4, 5, 6, 7

(b) 2, 1, 4, 3, 6, 5, 7

(c) 1, 3, 2, 5, 4, 7, 6

(d) 2, 3, 4, 5, 6, 7, 1

75. 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

76. 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

77. 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) All of the mentioned

78. 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) None of the mentioned

79. 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) All of the mentioned

80. 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) None of the mentioned

81. 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) None of the mentioned

82. Which of the following sorting algorithms does not have a worst case running time of $O(n^2)$?

(a) Insertion sort (b) Merge sort

(c) Quick sort (d) Bubble sort

83. For a linear search in an array of n elements the time complexity for best, worst and average case are....,.... and .... respectively.

(a) O(n), O(1), and O(n/2)

(b) O(1), O(n) and O(n/2)

(c) O/1, O(n) and O(n)

(d) O(1), O(n) and $\left(\frac{n-1}{2}\right)$

84. Using the standard algorithm, what is the time required to determine that a number n is prime?

(a) Linear time (b) Logarithmic time

(c) Constant time (d) Quadratic time

85. Average successful search time for sequential search on 'n' items is.

(a) $\frac{n}{2}$ (b) $\frac{(n-1)}{2}$

(c) $\frac{(n+1)}{2}$ (d) None of these

86. A text is made up of the characters a, b, c, d, e each occurring with the probability 0.12, 0.4, 0.15, 0.08 and .25 respectively. The optimal coding technique will have the average length of.

(a) 2.15 (b) 3.78

(c) 2.78 (d) 1.78

87. Suppose DATA array contains 1000000 elements. Using the binary search algorithm, one requires only about n comparisons to find the location of an item in the DATA array, then n is.

(a) 60 (b) 45

(c) 20 (d) none of these

88. 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

(a) full binary tree (b) 2-tree

(c) threaded tree (d) complete binary tree

89. 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) Preorder (b) Postorder

(c) Inorder (d) None of these

90. If each node in a tree has value greater than every value in its left subtree and has value less than every value in its right subtree, the tree is called.

(a) complete tree (b) full binary tree

(c) binary search tree (d) threaded tree

91. Which of the following sorting procedure is the slowest?

(a) Quick sort (b) Heap sort

(c) Shell sort (d) Bubble sort

92. A complete binary tree with the property that the value at each node is at least as large as the values at its children is called.

(a) Binary search tree

(b) AVL tree

(c) Completely balanced tree

(d) Heap

93. Which of the following best describes sorting?

(a) Accessing and processing each record exactly once

(b) Finding the location of the record with a given key

(c) Arranging the data (record) in some given order

(d) Adding a new record to the data structure

94. A characteristic of the data that binary search uses but the linear search ignores, is

(a) order of the list

(b) length of the list

(c) maximum value in the list

(d) mean of data values

95. A sort which compares adjacent elements in a list and switches where necessary is

(a) insertion sort (b) heap sort

(c) quick sort (d) bubble sort

96. A sort which iteratively passes through a list to exchange the first element with any element less than it and then repeats with a new first element is called.

(a) insertion sort (b) selection sort

(c) heap sort (d) quick sort

97. A full binary tree with n leaves contains

(a) n nodes (b) $\log_2 n$ nodes

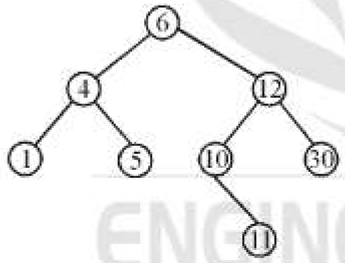
(c) $2n - 1$ (d) 2^n nodes

98. A full binary tree with n non-leaf nodes contains.

(a) $\log_2$ n nodes (b) n + 1 nodes

(c) 2n nodes (d) 2n + 1 nodes

99. Consider the tree shown in the figure below.



If this tree is used for sorting, then a new number 8, should be placed as the

(a) left child of node labelled 30

(b) right child of node labelled 5

(c) right child of node labelled 30

(d) left child of node labelled 10

100. The number of nodes in a complete binary tree of level 5 is.

(a) 15 (b) 25

(c) 63 (d) 71

101. 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) 11

(c) 12 (d) 13

102. 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

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

(a) $\log_2 n$ (b) 2n

(c) n (d) 2^n

104. A complete binary tree of level 5 has how many nodes?

(a) 15 (b) 25

(c) 63 (d) 33

105. Which of the following statements is used in the binary search algorithm to halve the array?

(a) middle Sub = (start Sub + stop Sub)/2;

(b) middle Sub = start Sub + stop Sub/2;

(c) middle Sub = middle Sub/2;

(d) middle Sub = (stop Sub - start Sub)/2;

106. The data for which you are searching is called.

(a) search argument

(b) sorting argument

(c) deletion argument

(d) binary argument

107. The maximum number of nodes of level i of a binary tree is.

(a) 2^{i-1} (b) 3^{i-1}

(c) i + 1 (d) 2^{i+2}

108. The maximum number of nodes in a binary tree of depth k is.

(a) 2^{k-1} (b) 3^{k-2}

(c) 2^{k-1} (d) $2^k + 2^x + 1$.

109. Which of the following is false?

(a) A binary search begins with the middle element in the array

(b) A binary search continues having the array either until a match is found or until there are no more elements to search.

(c) If the search argument is greater than the value located in the middle of the binary, the binary search continues in the lower half of the array.

(d) For a binary search to work, the data in the array must be arranged in either alphabetical or numerical order.

110. Identify the correct about a AVL tree.

I. In this tree heights of two subtrees of every node never differ by more than 1.

II. Balance factor of each node is –1, 0, 1.

III. Maximum height of a balanced binary search tree is 1.44 $\log_2^4$.

(a) I and II (b) II and III

(c) I and III (d) All of these

111. Traversing a binary tree first root and then left and right subtrees called ____ traversal.

(a) postorder (b) preorder

(c) inorder (d) none of these

112. A binary tree having n nodes and depth d will be about complete binary tree if

(a) any node nd at level less than d-1 has two sons

(b) it contains $2^{d+1} - 1$ nodes

(c) for any node nd in the tree with a right descendent at level d, nd must have a left son

(d) all of these

113. Which of the following statements are correct?

I. If each tree node contains a father field, then it's not necessary to use either stack or threads.

II. Traversal using father pointers is more time efficient than traversal of a threaded tree.

III. A in-threaded binary tree is defined as binary tree that is both left-in threaded and right-in threaded.

(a) II, and III (b) I and III

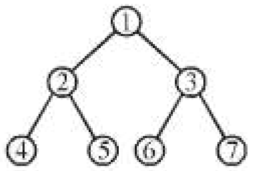
(c) All of these (d) None of these

114. The smallest number of key that will force a B-tree of order 3 to have a height 3 is.

(a) 12 (b) 10

(c) 7 (d) none of these

115. Consider the following tree:

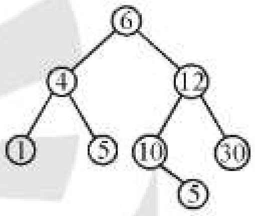


If the post order traversal gives ab – cd* + then the label of the nodes 1, 2, 3, ... will.

(a) +, –, *, a, b, c, d (b) a, –, b, +, c, *, d

(c) a, b, c, d, –, *, + (d) –, a, b, +, *, c, d

116. Consider the following tree:



If this tree is used for sorting, then a new number 8 should be placed as the

(a) left child of node labelled 30

(b) right child of node labelled 5

(c) right child of node labelled 30

(d) left child of node labelled 10

117. Number of possible ordered trees with 3 nodes A, B, C is.

(a) 16 (b) 12

(c) 13 (d) 14

118. 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 have more than 19 nodes

(b) has exactly 19 nodes

(c) has exactly 17 nodes

(d) cannot have more than 17 nodes

119. Average successful search time taken by binary search on a sorted array of 10 items is.

(a) 2.6 (b) 2.7

(c) 2.8 (d) 2.9

120. Number of possible binary trees with 3 nodes is

(a) 12 (b) 13

(c) 14 (d) 15

121. A-2-3 tree is a tree such that

1. All internal nodes have either 2 or 3 children.
2. All paths from root to the leaves have the same length.

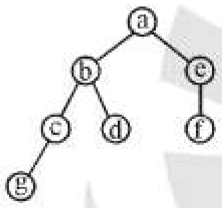
The number of internal nodes of a 2-3 tree having 9 leaves could be.

(a) 4 (b) 5

(c) 6 (d) 7

Common Data 122-123

A balanced tree is given below.



122. How many nodes will be come unbalanced when a node is inserted as a child of node g?

(a) 1 (b) 3

(c) 7 (d) 8

123. Which of the following sequences denotes the post order traversal sequence of the given tree?

(a) f eg cd ba (b) g c b dafe

(c) g c d b f e a (d) f ed g c b a

124. The minimum number of inter changes needed to convert the array 89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70 into a heap with maximum element at the root is.

(a) 1 (b) 2

(c) 3 (d) none of these

125. A binary tree is generated by inserting in order the following integers:

50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24

The number of nodes in the left subtree of the root respectively is.

(a) (4, 7) (b) (7, 4)

(c) (8, 3) (d) (3, 8)

126. Consider the following nested representation of binary trees (xyz) indicates y and z are the left right subtrees, respectively, of node x. Note that y and z may be NULL or further nested. Which of the following represents a valid binary tree?

(a) (1 2 (4 5 6 7)) (b) (1 ((2 3 4) 5 6) 7)

(c) (1 (234) (567)) (d) (1 (2 3 NULL) (4 5))

127. A full binary tree with n non-leaf nodes contains

(a) $\log_2$ n nodes (b) n + 1 nodes

(c) 2n nodes (d) 2n + 1 nodes

128. A full binary tree with n leaves contains.

(a) n nodes (b) $\log_2$ n nodes

(c) 2n – 1 nodes (d) 2n nodes

129. In a max-heap, element with the greatest key is always in the which node?

(a) Leaf node

(b) First node of left sub tree

(c) root node

(d) First node of right sub tree

130. Heap exhibits the property of a binary tree?

(a) True (b) False

131. What is the complexity of adding an element to the heap.

(a) O(log n)

(b) O(h)

(c) O(log n) & O(h)

(d) None of the mentioned

132. The worst case complexity of deleting any arbitrary node value element from heap is

(a) O(logn)

(b) O(n)

(c) O(nlogn)

(d) O(n2)

133. Heap can be used as ________________

(a) Priority queue

(b) Stack

(c) A decreasing order array

(d) None of the mentioned

134. If we implement heap as min-heap , deleting root node (value 1)from the heap. What would be the value of root node after second iteration if leaf node (value 100) is chosen to replace the root at start.

(a) 2 (b) 100

(c) 17 (d) 3

135. If we implement heap as maximum heap , adding a new node of value 15 to the left most node of right subtree . What value will be at leaf nodes of the right subtree of the heap.

(a) 15 and 1 (b) 25 and 1

(c) 3 and 1 (d) 2 and 3

136. An array consist of n elements. We want to create a heap using the elements. The time complexity of building a heap will be in order of

(a) O(n*n*logn) (b) O(n*logn)

(c) O(n*n) (d) O(n*logn*logn)

137. What is the space complexity of searching in a heap?

(a) O(logn) (b) O(n)

(c) O(1) (d) O(nlogn)

138. What is the best case complexity in building a heap?

(a) O(nlogn) (b) O(n2)

(c) O(n*longn *logn) (d) O(n)

139. Given the code ,choose the correct option that is consistent with the code

```
build(A,i)
left->2*i
right->2*i +1
temp-> i
if(left<=heap_length[A]ans A[left]>A[temp])
temp-> left
if(right =heap_length[A] and A[right]> A[temp])
temp->right
if temp!= i
swap(A[i],A[temp])
build(A,temp)
```

Here A is the heap

(a) It is the build function of max heap

(b) It is the build function of min heap

(c) It is general build function of any heap

(d) None of the mentioned

140. What is the location of parent node for any arbitary node i?

(a) (i/2) position

(b) (i+1)/ position

(c) floor(i/2) position

(d) ceil(i/2) position

141. Given an array of element 5,7,9,1,3,10,8,4. Tick all the correct sequences of elements after inserting all the elements in a min-heap.

(a) 1,3,4,7,8,9,10 (b) 1,4,3,8,9,5,7,10

(c) 1,3,4,5,8,7,9,10 (d) None of the mentioned

142 For construction of a binary heap with property that parent node has value less than child node. In reference to that which line is incorrect. Line indexed from 1.

```
add(int k)
{
heap_size++;
int i =heap_size-1;
harr[i]= k;
while(i !=0&&harr[parent(i)]<harr[i])
{
swap(&harr[i], &harr[parent(i)]);
i =parent(i);
}
}
```

(a) Line -3 (b) Line - 5

(c) Line - 6 (d) Line -7

143. Which of the following statements for a simple graph is correct?

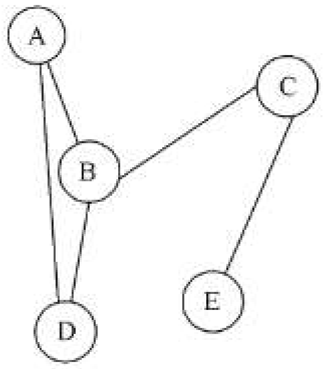
(a) Every path is a trail

(b) Every trail is a path

(c) Every trail is a path as well as every path is a trail

(d) None of the mentioned

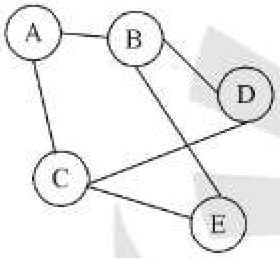
144. In the given graph identify the cut vertices.



(a) B and E (b) C and D

(c) A and E (d) C and B

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



(a) G is a complete graph

(b) G is not a connected graph

(c) The vertex connectivity of the graph is 2

(d) The edge connectivity of the graph is 1

146. What is the number of edges present in a complete graph having n vertices?

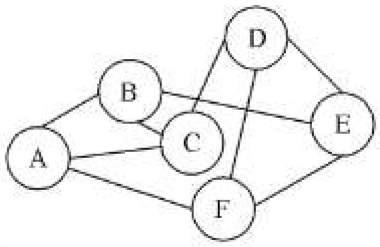
(a) (n*(n+1))/2

(b) (n*(n-1))/2

(c) n

(d) Information given is insufficient

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



(a) True

(b) False

148. A connected planar graph having 6 vertices, 7 edges contains _____________ regions.

(a) 15 (b) 3

(c) 1 (d) 11

149. 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) All of the mentioned

150. What is the maximum number of edges in a bipartite graph having 10 vertices?

(a) 24 (b) 21

(c) 25 (d) 16

151. 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

(b) A graph may contain no edges and no vertices

(d) None of the mentioned

152. 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) None of the mentioned

153. 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

154. A graph with all vertices having equal degree is known as a __________

(a) Multi Graph

(b) Regular Graph

(c) Simple Graph

(d) Complete Graph

155. 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) None of the mentioned

156. The number of elements in the adjacency matrix of a graph having 7 vertices is __________

(a) 7 (b) 14

(c) 36 (d) 49

157. Adjacency matrix of all graphs are symmetric.

(a) False (b) True

158. The time complexity to calculate the number of edges in a graph whose information in stored in form of an adjacency matrix is ____________

(a) O(V) (b) O(E2)

(c) O(E) (d) O(V2)

159 For the adjacency matrix of a directed graph the row sum is the _________ degree and the column sum is the ________ degree.

(a) in, out (b) out, in

(c) in, total (d) total, out

160. What is the maximum number of possible non zero values in an adjacency matrix of a simple graph with n vertices?

(a) (n*(n-1))/2

(b) (n*(n+1))/2

(c) n*(n-1)

(d) n*(n+1)

161. On which of the following statements does the time complexity of checking if an edge exists between two particular vertices is not, depends?

(a) Depends on the number of edges

(b) Depends on the number of vertices

(c) Is independent of both the number of edges and vertices

(d) It depends on both the number of edges and vertices

162. Which of these adjacency matrices represents a simple graph?

(a) [[1, 0, 0], [0, 1, 0], [0, 1, 1]]

(b) [[1, 1, 1], [1, 1, 1], [1, 1, 1]]

(c) [[0, 0, 1], [0, 0, 0], [0, 0, 1]]

(d) [[0, 0, 1], [1, 0, 1], [1, 0, 0]]

163. Given an adjacency matrix A = [[0, 1, 1], [1, 0, 1], [1, 1, 0]], how many ways are there in which a vertex can walk to itself using 2 edges.

(a) 2 (b) 4

(c) 6 (d) 8

164. If A[x+3][y+5] represents an adjacency matrix, which of these could be the value of x and y.

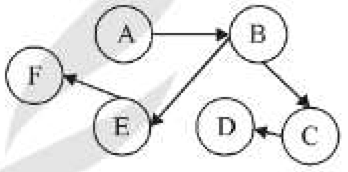
(a) x=5, y=3 (b) x=3, y=5

(c) x=3, y=3 (d) x=5, y=5

165. Every Directed Acyclic Graph has at least one sink vertex.

(a) True (b) False

166. Which of the following is a topological sorting of the given graph?



(a) A B C D E F (b) A B F E D C

(c) A B E C F D (d) All of the Mentioned

167. The topological sorting of any DAG can be done in _______ time.

(a) cubic (b) quadratic

(c) linear (d) logarithmic

168. If there are more than 1 topological sorting of a DAG is possible, which of the following is true.

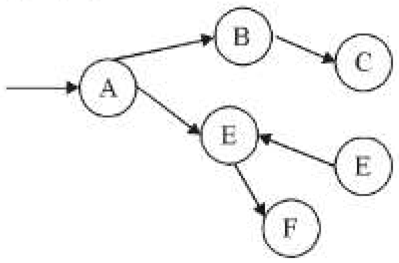
(a) Many Hamiltonian paths are possible

(b) No Hamiltonian path is possible

(c) Exactly 1 Hamiltonian path is possible

(d) Given information is insufficient to comment anything

169. What sequence would the BFS traversal of the given graph yield?



(a) A F D B C E

(b) C B A F D

(c) A B D C F

(d) F D C B A

170. Which of the given statement is true?

(a) All the Cyclic Directed Graphs have topological sortings

(b) All the Acyclic Directed Graphs have topological sortings

(c) All Directed Graphs have topological sortings

(d) None of the given statements is true

171. For any two different vertices u and v of an Acyclic Directed Graph if v is reachable from u, u is also reachable from v?

(a) True

(b) False

172. What is the value of the sum of the minimum in-degree and maximum out-degree of an Directed Acyclic Graph?

(a) Depends on a Graph

(b) Will always be zero

(c) Will always be greater than zero

(d) May be zero or greater than zero

173. In which of the following case(s) is it possible to obtain different results for call-by-reference and call-by-name parameter passing?

(a) Passing an expression as a parameter

(b) Passing an array as a parameter

(c) Passing a pointer as a parameter

(d) Passing an array element as a parameter

174. Match the pairs in the following:

List-I	
A.	Pointer data type
B.	Activation record
C.	Repeat-until
D.	Coercion

List-II	
P.	Type conversion
Q.	Dynamic data structure
R.	Recursion
S.	Nondeterministic loop

175. Indicate all the true statements from the following:

(a) A programming language not supporting either recursion or pointer type does not need the support of dynamic memory allocation.

(b) Although C does not support call by name parameter passing, the effect can be correctly simulated in C.

176. What does the following code do?

```
var a, b : integer;
begin
    a:=a+b;
    b:=a–b;
    a:=a–b
end;
```

(a) exchanges (a) and (b)

(b) doubles (a) and stores in (b)

(c) doubles (b) and stores in (a)

(d) leaves (a) and (b) unchanged

177. In which one of the following cases is it possible to obtain different results for call-by reference and call-by-name parameter passing methods?

(a) Passing a constant value as a parameter

(b) Passing the address of an array as a parameter

(c) Passing an array element as a parameter

(d) Passing an array

178. Consider the following C function definition

```
int Trial (int a, int b, int c)
{
    if ((a > = b) && (c < b) return b;
    else if (a > = b) return Trial (a,c,b);
    else return Trial (b,a,c);
}
```

The function Trial:

(a) Finds the maximum of a, b and c

(b) Finds the minimum of a, b and c

(c) Finds the middle number of a, b and c

(d) None of the above

179. The following C declarations

```
struct node{
    int i;
    float j;
};
struct node *s[10];
```

define s to be

(a) An array, each element of which is a pointer to a structure of type node

(b) A structure of 2 fields, each field being a pointer to an array of 10 elements

(c) A structure of 3 fields: an integer, a float, and an array of 10 elements

(d) An array, each element of which is a structure of type node

180. The most appropriate matching for the following pairs

List-I

X. m=malloc(5); m = NULL;

Y. free(n); n->value=5;

Z. char *p; *p ='a';

List-II

1. using dangling pointers

2. using uninitialized pointers

3. lost memory Codes:

(a) X–1, Y–3, Z–2

(b) X–2, Y–1, Z–3

(c) X–3, Y–2, Z–1

(d) X–3, Y–1, Z–2

181. The value of j at the end of the execution of the following C program is______

```
int incr (int i) {
    static int count =0;
    count = count + i;
    return (count);
}
main () {
    int i,j;
    for (i = 0; i <=4; i++)
    j = incr(i);
}
```

(a) 10 (b) 4

(c) 6 (d) 7

182. The results returned by function under value-result and reference parameter passing conventions

(a) Do not differ

(b) Differ in the presence of loops

(c) Differ in all cases

(d) May differ in the presence of exception

183. Consider the following C function

```
int f (int n)
{
    static int i = 1;
    if (n > = 5) return n ;
    n = n + i ;
    i ++;
    return f (n);
}
```

The value returned by f(1) is

(a) 5 (b) 6

(c) 7 (d) 8

184. Consider the following C program segment:

```
    char p [20] ;
    char * s = "string" ;
    int length = strlen (s);
    for (i = 0 ; i < length; i++)
    p[i] = s [length-i];
printf ("%s", p);
```

The output of the program is

(a) gnirts

(b) string

(c) gnirt

(d) no output is printed

185. Which of the following are essential features of an object-oriented programming languages?

1. Abstraction and encapsulation
2. Strictly-typedness
3. Type-safe property coupled with sub-type rule
4. Polymorphism in the presence of inheritance

(a) 1 and 2 only

(b) 1 and 4 only

(c) 1, 2 and 4 only

(d) 1, 3 and 4 only

186. Consider the following C-program

```
void foo (int n, int sum) {
    int k = 0, j = 0;
    if (n = = 0) return;
    k = n % 10;
    j = n /10;
    sum = sum + k;
    foo (j, sum);
    printf ("% d", k);
    }
int main 0 {
    int a = 2048, sum = 0;
    foo (a, sum);
    printf ("%d\n", sum);
}
```

What does the above program print?

(a) 8, 4, 0, 2, 14

(b) 8, 4, 0, 2, 0

(c) 2, 0, 4, 8, 14

(d) 2, 0, 4, 8, 0

187. What is the output printed by the following program?

```
# include <stdio.h>
int f(int n, int k)
{
    if (n = = 0) return 0;
    else if (n% 2) return f(n/2, 2*k)+k;
    else return f(n/2, 2*k) – k;
}
int main()
{
    printf("%d", f(20, 1));
    return 0;
}
```

(a) 5 (b) 8

(c) 9 (d) 20

188. Consider the following C function:

```
int f(int n)
{   static int r = 0;
    if(n < = 0) return 1;
    if(n > 3)
    {
        r = n;
        return f(n – 2) + 2;
    }
    return f(n —1) + r;
}
```

What is the value of f(5)?

(a) 5 (b) 7

(c) 9 (d) 18

189. What is printed by the following C program?

```
int f(int x, int *py, int ** ppz)
{   inty, z;
    **ppz + = 1; z = *ppz;
    *py + = 2; y = *py;
    x + = 3;
    return x + y + z;
}
void main()
{   int c, *b, **a;
    c = 4; b = &c; a = & b;
    printf ("% d", f (c, b, a));
```

(a) 18

(b) 19

(c) 21

(d) 22

190. What does the following program print?

```
#include < stdio.h>
void f(int *p, int *q)
{
    p = q;
    *p = 2;
}
int i = 0, j = 1;
int main()
{
    f(&i, &j);
    printf ("%d %d\n", i, j);
    return 0;
}
```

(a) 2 2

(b) 2 1

(c) 0 1

(d) 0 2

191. What is the value printed by the following C program?

```
#include < stdio.h>
int f(int *a, int n)
{
    if (n <= 0) return 0;
    else if (*a % 2 == 0)
        return *a + f(a + 1, n - 1);
    else return *a - f(a + 1, n - 1);
}
int main ()
{
    int a [] = {12, 7, 13, 4, 11, 6};
    printf("%d", f(a, 6));
    return 0;
}
```

(a) –9

(b) 5

(c) 15

(d) 19

192. What will be the output of the following C program segment?

```
char inChar = 'A';
switch(inChar)
{
    case 'A' : printf ("Choice A\ n");
    case 'B' :
    case 'C' : printf ("Choice B");
    case 'D' :
    case 'E' :
    default : printf ("NO Choice");}
```

(a) No Choice

(b) Choice A

(c) Choice A, Choice B No Choice

(d) Program gives no output as it is erroneous

193. The output of the following C program is_____.

```
void f1 (int a, int b) {
    int c;
    c=a; a=b; b=c;
}
void f2 (int *a, int *b) {
    int c;
    c=*a; *a=*b; *b=c;
}
int main() {
    int a=4, b=5, c=6;
    f1 (a, b);
    f2 (&b, &c);
    printf ("%d", c - a - b);
}
```

194. Consider the following C program segment.

```
#include <stdio.h>
int main() {
    char s1[7] = "1234", *p;
    p = s1 + 2;
    *p = '0' ;
    printf ("%s", si);
}
```

What will be printed by the program?

(a) 12

(b) 120400

(c) 1204

(d) 1034

195. Consider the following C program.

```
# include < stdio.h>
int main() {
    static int a[ ] = {10, 20, 30, 40, 50};
    static int *p[ ] = {a, a+3, a+4, a+1, a+2};
    int **ptr = p; ptr++;
    printf ("%d%d", prt-p, **ptr);
}
```

The output of the program is______.

196. The value printed by the following program is ________.

```
void f(int* p, int m)
{
    m = m + 5;
    *p = *p + m;
    return;
}
void main ()
{
    int i = 5, j = 10;
    f(&i, j);
    printf("%d", i+j);
}
```

197. Consider the following C program.

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

void printlength (char *s, char *t)
{   unsigned int c = 0;
    int len = ((strlen (s) - strlen (t)) > c) ?
strlen                                       (s) :
strlen (t);
    printf("%d\n", len);
}
void main ()
{   char *x = "abc";
    char *y = "defgh";
    printlength (x,y);
}
```

Recall that strlen is defined in string.h as returning a value of type size_t, which is an unsigned int. The output of the program is _________.

198. Consider the following two functions.

```
void fun 1(int n)          void fun2 (int n)
{                          {
    if (n ==0)return;     if(n ==0) return;
    printf("%d", n);        printf ("%d", n);
    fun2(n – 2);            fun 1 (++n);
    printf("%d", n);        print f("%d", n);
}                          }
```

The output printed when funl (5) is called is

(a) 53423122233445

(b) 53423120112233

(c) 53423122132435

(d) 53423120213243

199. Consider the following function implemented in C:

```
void printxy (int x, int y)
{
    int *ptr;
    x = 0;
    ptr = &x;
    y = *ptr;
    *ptr = 1;
    Printf("%d, %d", x, y);
}
```

The output of invoking printxy (1, 1) is

(a) 0, 0

(b) 0, 1

(c) 1, 0

(d) 1, 1

200. Consdier the following C program:

```
#include<stdio.h >
void fun1(char *s1, char *s2)
    {
    char *tmp;
    tmp = s1 ;
    s1 = s2;
    s2 = tmp;
    }
void fun2 (char **s1, char **s2)
    {
    char *tmp;
    tmp = s1 ;
    *s1 = *s2;
    *s2 = tmp;
    }
int main ( )
{
char *str1 = "Hi", *str2 = "Bye";
fun1 (str1,str2);
 printf ("%s%s", str1,str2);
fun2 ( &str1, &str2);
 printf ("%s%s", str1,str2);
return 0;
}
```

The output of teh plrogram above is

(a) Hi Bye Bye Hi

(b) Hi Bye Hi Bye

(c) Bye Hi Hi Bye

(d) Bye Hi Bye Hi

201. Consider the following C program:

```
#include <stdio.h>
int main( )
{
    int arr[ ] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2,
5}, *ip = arr + 4;
    printf("%d\n", ip[1]);
    return 0;
}
```

The number that will be displayed on execution of the program is _________.

202. Consider the following C program:

```
#include <stdio.h>
int main ( )
{
    int a[ ] = { 2, 4, 6, 8, 10} ;
    int i, sum = 0, *b = a + 4 ;
    for (i = 0; i < 5; i + +)
    sum = sum + (*b–i) – *(b–i) ;
    printf ("%d\n", sum) ;
return 0;
}
```

The output of the above C program is_________.

❍❍❍

Answer Key

1. *Ans. (b)*
2. *Ans. (a)*
3. *Ans. (d)*
4. *Ans. (c)*
5. *Ans. (b)*
6. *Ans. (c)*
7. *Ans.* 1898320 1898320 1898324 1898340
8. *Ans. 4*
9. *Ans. (a)*
10. *Ans. (b)*
11. *Ans. (b)*
12. *Ans. (c)*
13. *Ans. (d)*
14. *Ans. (d)*
15. *Ans. (d)*

    Since there are 15 int elements and each int is of 4bytes, we get 15*4 = 60bytes
16. *Ans. (b)*
17. *Ans. (c)*
18. *Ans. (b)*
19. *Ans. (d)*
20. *Ans. (c)*
21. *Ans. (a)*
22. *Ans. (a)*
23. *Ans. (a)*
24. *Ans. (d)*
25. *Ans. (c)*
26. *Ans. (d)*
27. *Ans. (b)*
28. *Ans. (a)*
29. *Ans. (b)*
30. *Ans. (c)*
31. *Ans. (a)*
32. *Ans. (c)*
33. *Ans. (a)*
34. *Ans. (a)*
35. *Ans. (a)*
36. *Ans. (a)*
37. *Ans. (c)*
38. *Ans. (a)*
39. *Ans. (b)*
40. *Ans. (a)*
41. *Ans. (b)*
42. *Ans. (a)*
43. *Ans. (d)*
44. *Ans. (a)*
45. *Ans. (c)*
46. *Ans. (b)*
47. *Ans. (d)*
48. *Ans. (b)*
49. *Ans. (c)*
50. *Ans. (b)*
51. *Ans. (a)*
52. *Ans. (a)*
53. *Ans. (b)*
54. *Ans. (c)*
55. *Ans. (a)*
56. *Ans. (a)*
57. *Ans. (a)*
58. *Ans. (b)*
59. *Ans. (d)*
60. *Ans. (d)*
61. *Ans. (a)*
62. *Ans. (a)*
63. *Ans. (b)*
64. *Ans. (c)*
65. *Ans. (c)*
66. *Ans. (c)*
67. *Ans. (d)*
68. *Ans. (d)*
69. *Ans. (a)*
70. *Ans. (a)*

71. *Ans. (c)*

72. *Ans. (d)*

73. *Ans. (d)*

74. *Ans. (b)*

75. *Ans. (d)*

76. *Ans. (a)*

77. *Ans. (c)*

78. *Ans. (a)*

79. *Ans. (c)*

80. *Ans. (b)*

81. *Ans. (b)*

82. *Ans. (b)*

83. *Ans. (c)*

84. *Ans. (a)*

85. *Ans. (c)*

86. *Ans. (a)*

87. *Ans. (c)*

88. *Ans. (d)*

89. *Ans. (d)*

90. *Ans. (c)*

91. *Ans. (d)*

92. *Ans. (d)*

93. *Ans. (c)*

94. *Ans. (a)*

95. *Ans. (d)*

96. *Ans. (b)*

97. *Ans. (c)*

98. *Ans. (d)*

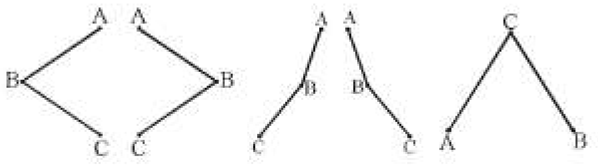
99. *Ans. (d)*

100. *Ans. (c)*

101. *Ans. (d)*

102. *Ans. (d)*

The 5 binary trees are



103. *Ans. (d)*

It can be proved by induction that a strictly binary tree with 'n' leaf nodes will have a total of 2n – 1 nodes.

∴ Number of non-leaf nodes = (2n – 1) – n = n – 1.

104. *Ans. (c)*

105. *Ans. (b)*

106. *Ans. (a)*

107. *Ans. (a)*

108. *Ans. (a)*

109. *Ans. (b)*

110. *Ans. (b)*

111. *Ans. (b)*

112. *Ans. (a)*

113. *Ans. (b)*

114. *Ans. (c)*

115. *Ans. (a)*

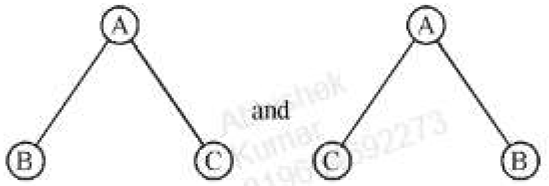
116. *Ans. (d)*

If it is to be used for sorting, label of left child should be less than the label of the current node.

So, coming down the tree, we get left child of node labelled 10 as the correct slot for 8.

117. *Ans. (b)*

The tree may be of depth 2 or 1. If 2, we have 6 possible trees. Because one of the three nodes A, B, C may be the root. The next level may be one of the remaining two. So, 6. If the depth is 2, the root may be one of the 3 nodes A, B, C corresponding to a root say A, two trees are possible as this.



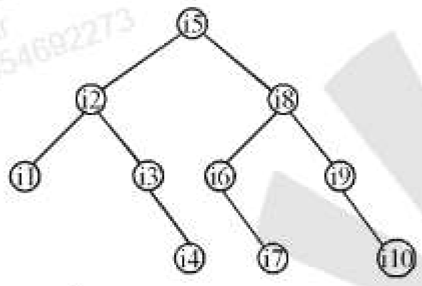
So, 6 possibilities. So, altogether 12 possible trees.

118. ***Ans. (b)***

A strictly binary tree with 'n' leaves must have (2n – 1) nodes. Verify for some small 'n'. This can be proved by the principle of mathematical induction.

119. ***Ans. (d)***

The 10 items i1, i2......i10 may be arranged in a binary search tree as shown in the figure below. So, to match i5, the number of comparison needed is 1; for i2, it is 2, for i8 it is 2, for i1 it is 3 and so on.

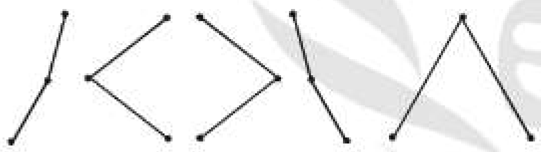


∴ Average

$$= \frac{(1+(2+2)+(3+3+3+3)+(4+4+4))}{10} = 2.9$$

120. ***Ans. (d)***

The five possible trees are.



121. ***Ans. (a)***

122. ***Ans. (b)***

123. ***Ans. (c)***

124. ***Ans. (b)***

125. ***Ans. (b)***

126. ***Ans. (c)***

127. ***Ans. (d)***

128. ***Ans. (c)***

129. ***Ans. (c)***

130. ***Ans. (a)***

131. ***Ans. (c)***

132. ***Ans. (a)***

133. ***Ans. (a)***

134. ***Ans. (a)***

135. ***Ans. (a)***

136. ***Ans. (b)***

137. ***Ans. (b)***

138. ***Ans. (d)***

139. ***Ans. (a)***

140. ***Ans. (c)***

141. ***Ans. (a)***

142. ***Ans. (b)***

143. ***Ans. (a)***

144. ***Ans. (d)***

145. ***Ans. (c)***

146. ***Ans. (b)***

147. ***Ans. (b)***

148. ***Ans. (b)***

149. ***Ans. (a)***

150. ***Ans. (c)***

151. ***Ans. (b)***

152. ***Ans. (b)***

153. ***Ans. (a)***

154. ***Ans. (b)***

155. ***Ans. (c)***

156. ***Ans. (d)***

157. ***Ans. (a)***

158. ***Ans. (d)***

159. ***Ans. (b)***

160. ***Ans. (c)***

161. ***Ans. (c)***

162. ***Ans. (d)***

163. ***Ans. (c)***

164. ***Ans. (a)***

165. ***Ans. (a)***

166. ***Ans. (b)***

167. ***Ans. (c)***

168. ***Ans. (b)***

169. ***Ans. (c)***

170. ***Ans. (a)***

171. ***Ans. (b)***

172. ***Ans. (b)***

173. ***Ans. (d)***

Passing an array element by call-by-name behaves like call-by-value. Therefore, there may be different results for call-by-reference and call-by-name parameter passing for passing an array elements.

174. ***Sol.***

- In programming language, coercion is a way of implicitly or explicity, changing an entity of one datatype into another.
- Repeat-until is a nondeterministic loop.
- The portion of the stack used for an invocation of a function is called the function's stack frame or activation record. So activation record is used for recursion.
- Dynamic data structure is used for pointer data type.

a-q, b-r, c-s, d-p

175. ***Sol.***

Both are correct.

176. ***Ans. (a)***

Given program is an example of swapping.

177. ***Ans. (c)***

Call by name and call by reference parameter passing techniques may differ when passes an array element as a parameter.

178. ***Ans. (d)***

Take some values and verify the options as 10, 6, 7.

179. ***Ans. (a)***

It defines an array each element of which is a pointer to a structure of type node.

180. ***Ans. (d)***

$$\left.\begin{array}{l}\text{m = malloc (S);}\\ \text{m = NULL;}\end{array}\right\}\rightarrow \text{Lost memory :}$$

(memory is lost for m)

$$\left.\begin{array}{l}\text{free(n);}\\ \text{n} \rightarrow \text{values = 5;}\end{array}\right\}\rightarrow \text{Dangling pointer.}$$

$$\left.\begin{array}{l}\text{char *p;}\\ \text{*p = 'a';}\end{array}\right\}\rightarrow \text{Uninitialized pointer.}$$

(p is not initialized yet. So trying for * p will fail)

181. ***Ans. (a)***

Count is static variable so every time when value is changed it becomes the new Value of count. So, first count value is 0 then 1, then 3, then 6 at last 10.

182. ***Ans. (b)***

Call by value result and call by reference may differ in the presence of aliasing (sending the same parameter more than once).

183. ***Ans. (c)***

```
1.  intf(int n)
2.  {       static int i = 1;
3.          if (n >= 5) return n;
4.          n = n + i;
5.          i ++;
6.          return f(n)
7.  }
```

	Iteration	n	i
Initialize	0	1	1
	1	1+1=2	2
	2	2+2=4	3
	3	4+3=7	4

So when we call f(1), the value i = 4 and n = 7, if condition is $7 \geq 5$ so returns the value 7.

184. ***Ans. (d)***

Because of first character is '\0' nothing will be printed.

185. ***Ans. (b)***

Object oriented programming language is

Object based PL+Abstraction+Inheritance.

The last two (abstraction and inheritance) are must for any L to be OOPL.

186. ***Ans. (d)***

```
void foo(int n, int sum){
    int k = 0, j = 0;
    if (n = = 0) return;
    k = n%10; j = n/10;
    sum = sum + k;
    foo(j, sum);
    printf("%d", k);
}
int main(){
    int a = 2048, sum = 0;
    foo(a, sum);
    printf("%d\n", sum);
}
```

The function foo is recursive function When we call foo(a, sum) = foo (2048, 0)

k	j	sum
$k = 2048\%10 = foo(204,8)$ foo(204,8)	$j = 204$	$sum = 0 + 8 = 8$
$k = 204\%10 = 4$ foo(20,12)	$j = 20$	$sum = 8 + 4 = 12$
$k = 20\%10 = 0$ foo(2,12)	$j = 2$	$sum = 12 + 0 = 12$
$k = 2\%10 = 2$	$j = 0$	$sum = 12 + 2 = 14$

foo(0, 14) function will be terminated and value of k will print in stack way i.e. 2, 0, 4, 8 and sum = 0. Since sum is a local variable in the main function so the print sequence is 2, 0, 4, 8, 0.

187. ***Ans. (c)***

The sequence has to be followed.

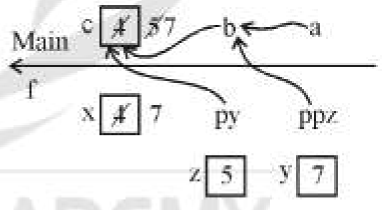
6. f(20, 1) = 9
5. f(10, 2) – 1 = 9
4. f(5, 4) – 2 = 10
3. f(2, 8) + 4 = 12
2. f(1, 16) – 8 = 8
1. f(0, 32) + 16 = 16

188. ***Ans. (d)***

$$f(3)+2=18 \qquad [(n>3): f(n-2)+2]$$
$$\uparrow$$
$$f(2)+5=16 \qquad [f(n-2)+r]$$
$$\uparrow$$
$$f(1)+5=11 \qquad [f(n-2)+r]$$
$$\uparrow$$
$$f(0)+5=6 \qquad [f(n-2)+r]$$
$$\uparrow$$
$$1 \qquad [n \le 0]$$

189. ***Ans. (b)***

f(c, b, a) is called by the main( ) function the graphical execution of the program is given below.



$\Rightarrow x + y + z = 7 + 7 + 5 = 19$

$\Rightarrow$ x + y + z = 7 + 7 + 5 = 19

int y, z;

**ppz = **ppz + 1;

z = *ppz = 5;

*py = *py + 2;

y = *py = 7

x = 4 + 3 = 7

return x + y + z = 19

190. ***Ans. (d)***

In the given program it begin from main function, i and j are globally initialized by 0 and 1. So when we call function f(& i, & j) the address of i and j are passed.

when $p = q$ and $*p = 2$ means

$*q = 2$, so value of $*q$ is passed,

and value of $*q$ return to j.

Value of i and j are 0 and 2 respected.

So printf("% d %d", i, j) gives output (0, 2)

191. ***Ans. (c)***

a[ ] =

12	7	13	4	11	6

So f(a, 6) → first this call

Thus is the number stored in the array and we can access these by using.

*a – f(a, 6)

↓ 12

So 12 + f(a + 1, n – 1) {even}

↓

7 – f(a + 1, n – 1) {odd}

↓

13 – f(a + 1, n – 1) {odd}

↓

4 + f(a + 1, n – 1) {even}

↓

11 – f(a + 1, n – 1) {odd}

↓

6 + f(a + 1, 0) {even}

↓

0

$= 12 + [7 – [13 – [4 + [11 – (6 + 0)]]]]$

$= 12 + [7 – [13 – [4 + [11 – 6]]]$

$= 12 + [7 – [13 – [4 + 5]]]$

$= 12 + (7 – (13 – 9))$

$= 12 + (7 – 4)$

$= 12 + 3 = 15$

192. ***Ans. (c)***

There is no break statement present in the code. Hence the switch case will jump to case: A and execute every statement from there.

∴ Choice A

Choice B No choice.

193. ***Sol.***

Here both f_1 and f_2 swaps the two arguments, f_1 is call by value but f_2 is called by reference hence changes made by only f_2 are reflected in main function.

After f_2 is called, the values of a, b and c are 4, 6 and 5 respectively.

Hence the output is $5 – 6 – 4 = –5$

194. ***Ans. (c)***

1	2	3	4	\0		
1000	1001	1002	1003	1004	1005	1006

s_1 1000

p = 1002

*p = '0'; ⇒

1	2	0	4	\0		
1000	1001	1002	1003	1004	1005	1006

Therefore the output is 1204.

195. ***Sol.***

a	10	20	30	40	50
	a+0	a+1	a+2	a+3	a+4

p	a	a+3	a+4	a+1	a+2
	p+0	p+1	p+2	p+3	p+4

ptr P

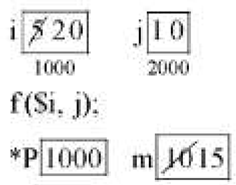
ptr ++ ; ⇒ ptr is p + 1.

ptr – p = (p + 1) – p = 1

ptr = *(*(p+1)) = *(a + 3) = **40

∴ Output is 1, 40 which is **140**.

196. ***Sol.***



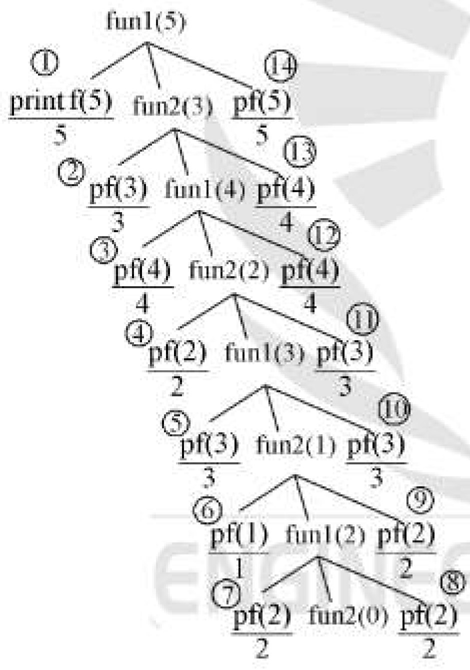
1. m = m + 5; $\because$ m = 15
2. *P = *P + m; $\because$ *P = 20
3. Print f(i + j); $\because$ 20 + 10 = 30

So value printed by above code is 30.

197. ***Ans. 3***

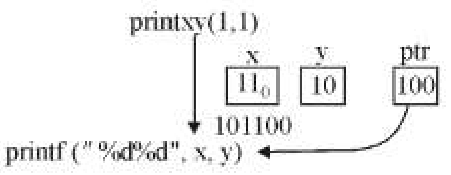
Strlen(s) – Strlen(t) will return an unsigned integer which is greater than 'c' so 'len' variable holds value '3'. So output is 3.

198. ***Ans. (a)***



output: 53423122233445

199. ***Ans. (c)***



Output = 1, 0

200. ***Ans (a)***

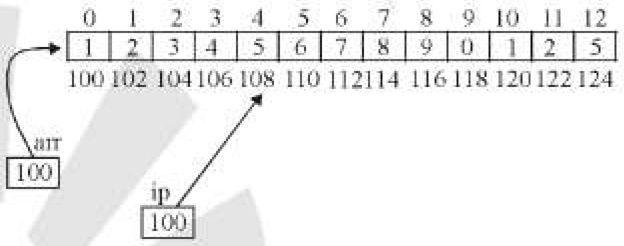
fun1 is call by value. So no change to original *str1 and *str2, so printf will print Hi Bye

fun2 is call by refference. So it will swap the value of str1 and str2. so, str1 points to "Bye" ans str2 points to "Hi".

So,printf will point Bye Hi.

Final output: Hi Bye Bye Hi

201. ***Ans. (6)***



ip = arr + 4

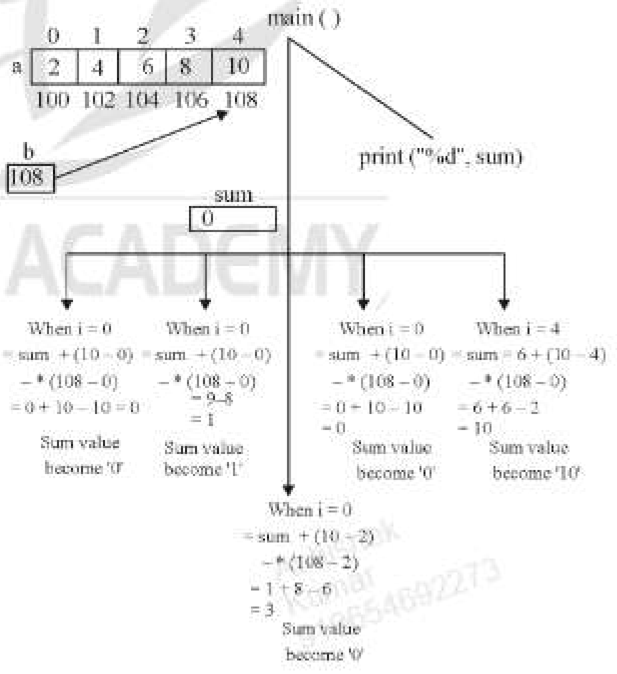
= 100 + (4 × 2)

= 108 / ip pointer pointing to 108

printf("%d", ip[1]);

output is 6

202. ***Ans. (10)***


main ( )
0 1 2 3 4
a 2 4 6 8 10
100 102 104 106 108
b
108
sum
0
print ("%d", sum)
When i = 0
= sum + (10 – 0) – * (108 – 0)
= 0 + 10 – 10 = 0
Sum value become '0'
When i = 0
= sum + (10 – 0) – * (108 – 0)
= 9–8 = 1
Sum value become '1'
When i = 0
= sum + (10 – 0) – * (108 – 0)
= 0 + 10 – 10 = 0
Sum value become '0'
When i = 4
= sum = 6 + (10 – 4) – * (108 – 0)
= 6 + 6 – 2 = 10
Sum value become '10'
When i = 0
= sum + (10 – 2) – * (108 – 2)
= 1 + 8 – 6 = 3
Sum value become '0'