# Uka Tarsadia University (Diwaliba Polytechnic)

## Diploma in Computer Engineering/Information Technology

## Objective Type Questions (Data and File Structures- CE1006)

### Unit-I Basic Concepts of Data Structures

1.Which of the following is an example of static memory allocation?

a) Linked list

b) Stack

c) Queues

d) Array

2. Array is preferred over linked list for the implementation of _____

a) Radix sort

b) Insertion sort

c) Binary search

d) Polynomial evaluation

3. How can we describe an array in the best possible way?

a) The Array shows a hierarchical structure.

b) Arrays are immutable.

c) Container that stores the elements of similar types

d) The Array is not a data structure

4._____ defines a set of primitive elements which do not involves any other element as its sub-part.

a) Linear Data Structure

b) Primitive Data Structure

c) Non-Primitive Data Structure

d) Non Linear Data Structure

5. Primitive data Structures are generally _____ data types in programming language.

a) Built in Data Types

b) User Defined Data Types

6. When an algorithm is written in the form of a programming language, it becomes a _____

a) Flowchart

b) Program

c) Pseudo code

d) Syntax

7. Any algorithm is a program.

a) True

b) False

8. Another name for 1-D arrays.

a) Linear arrays

b) Lists

c) Horizontal array

d) Vertical array

9. An algorithm is a _____ set of precise instructions for performing computation.

a) Infinite

b) Finite

c) Constant

d) None of the mentioned

10. Out of the following which property algorithms does not share?

a) Input

b) Finiteness

c) Generality

d) Constancy

## Unit-II Stack and Queue

11. Process of inserting an element in stack is called _____

a) Create

b) Push

c) Evaluation

d) Pop

12. Process of removing an element from stack is called _____

a) Create

b) Push

c) Evaluation

d) Pop

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

A+(B*C)

a) +A*CB

b) *B+AC

c) +A*BC

d) *A+CB

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

(A*B)+(C*D)

a) +*AB*CD

b) *+AB*CD

c) **AB+CD

d) +*BA*CD

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

A+B*C^D

a) +A*B^CD

b) +A^B*CD

c) *A+B^CD

d) ^A*B+CD

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

a) +

b) $

c) ^

d) &

17. Out of the following operators (|, *, +, &, $), the one having lowest priority is _____

a) +

b) $

c) |

d) &

18. 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)*+

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

a) True

b) False

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

a) True

b) False

21. What is the postfix expression for the infix expression?

a-b-c

a) -ab-c

b) ab – c –

c) – -abc

d) -ab-c

22. Which deletion can be insertion take place only at the other end(rear) and done from one end (front)?

a) linked list

b) Stack

c) Tree

d) queue

e)both a&c

23. For Breadth-First Traversal on a graph is the data structure required?

a) Stack

b) queue

c) array

d) Tree

e) Both a&b

24. Queue follows _____

a) FIFO (First In First Out) principle

b) LIFO (Last In First Out) principle

c) Linear tree

d) Ordered array

e) all of these

25. Circular Queue is also called _____

a) Square Buffer

b) Ring Buffer

c) Rectangle Buffer

d) Curve Buffer

e) None of these

26. In what order will they be removed If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time

a) ABCD

b) DCAB

c) DCBA

d) ABDC

e) All of the above

27. Which elements not in middle but can be inserted or deleted at/from both the ends?

a) Circular queue

b) Priority queue

c) Queue

d) DE queue

e) All of these

28. if implemented using an array of size MAX_SIZE, gets full when

a) Front = (rear + 1)mod MAX_SIZE

b) Front = rear + 1

c) Rear = MAX_SIZE – 1

d) Rear = front

e) None of above

29. Major role of queue server in _____

a) Simulation of heapsort

b) Simulation of arbitrary linked list

c) Simulation of limited resource allocation

d) Simulation of recursion

e) Both a&b

30. Which is not the type of queue?

a) Single ended queue

b) Ordinary queue

c) Circular queue

d) Priority queue

e) Both c&d

31. …… form of access is used to add and remove nodes from a queue.

a) LIFO, Last In First Out

b) FIFO, First In First Out

c) Both a and b

d) None of these

e) INFO

32. Which fields holds the elements of the stack.

a) INFO

b) TOP

c) LINK

d) NULL

e) None

33. Which form of access is used to add remove nodes from a stack?

a) LIFO

b) FIFO

c) Both A and B

d) None of these

e) INFO

34. Which pointer behaves as the top pointer variable of the stack.

a) Stop pointer

b) Begin pointer

c) avail pointer

d) start pointer

e) Both b&c

35. …… of the queue added a new nodes

a) Front

b) middle

c) back

d) Both A and B

e) None

36. A null pointer of the last node in the list signals …….

a) Beginning of the stack

b) Bottom of the stack

c) Middle of the stack

d) In between some value

e) Both a&b

37. The queue is a ……

a) FIFO

b) LIFO

c) LOFI

d) FILO

e) All of these

38. Which name does not relate to stacks?

a) FIFO lists

b) LIFO lists

c) Push down lists

d) Piles

e) both a&b

39. operation retrieval of items in a stack is ……

a) access

b) pop

c) retrieval

d) push

e) none

## Unit-III Linked List

40. Choose the statement which is incorrect with respect to dynamic memory allocation.

a) Memory is allocated in a less structured area of memory, known as heap

b) Used for unpredictable memory requirements

c) Execution of the program is faster than that of static memory allocation

d) Allocated memory can be changed during the run time of the program based on the requirement of the program

41. Which of the following header files must necessarily be included to use dynamic memory allocation functions?

a) stdlib.h

b) stdio.h

c) memory.h

d) dos.h

42. The type of linked list in which the node does not contain any pointer or reference to the previous node is _____

a) Circularly singly linked list

b) Singly linked list

c) Circular doubly linked list

d) Doubly linked list

43. The advantage of using linked lists over arrays is that _____

a) Linked list is an example of linear data structure

b) Insertion and deletion of an element can be done at any position in a linked list

c) Linked list can be used to store a collection of homogenous and heterogeneous data types

d) The size of a linked list is fixed

44. What is the return type of malloc() or calloc()?

a) int *

b) int **

c) void *

d) void **

45. Which function is used to delete the allocated memory space?

a) Dealloc()

b) free()

c) Both A and B

d) None of the above

46. Among 4 header files, which should be included to use the memory allocation functions like malloc(), calloc(), realloc() and free()?

a) #include<string.h>

b) #include<stdlib.h>

c) #include<memory.h>

d) Both b and c

47. Which of the following is/are true

a) calloc() allocates the memory and also initializes the allocates memory to zero, while memory allocated using malloc() has random data.

b) malloc() and memset() can be used to get the same effect as calloc()

c) Both malloc() and calloc() return 'void *' pointer

d) All of the above

48. Which of the following is true?

a) "ptr = calloc(m, n)" is equivalent to following

b) r = malloc(m * n);

c) "ptr = calloc(m, n)" is equivalent to following

d) r = malloc(m * n); memset(ptr, 0, m * n);

49. Which of the following statement is correct prototype of the malloc() function in c ?

a) int* malloc(int);

b) Char* malloc(char);

c) unsigned int* malloc(unsigned int);

d) void* malloc(size_t);

50. Specify the 2 library functions to dynamically allocate memory?

a) malloc() and memalloc()

b) alloc() and memalloc()

c) malloc() and calloc()

d) memalloc() and faralloc()

51. malloc() returns a float pointer if memory is allocated for storing float's and a double pointer if memory is allocated for storing double's.

a) TRUE

b) FALSE

c) May Be

d) Can't Say

52. What is the output of this program?

```
#include <stdio.h>
#include  <stdlib.h>
int main() {
  int i, numbers[1];
  numbers[0] = 15;
  free(numbers);
  printf("Stored integers are ");
  printf("numbers[%d] = %d ", 0, numbers[0]);
  return 0;
}
```

a) 15

b) Compilation error

c) 0

d) garbage value

53. What is the output of this program?

```
#include <stdio.h>
#include  <stdlib.h>
int main() {
   int *j = (int*)malloc(4 * sizeof(int));
```

```
  *j = 15;
  free(j);
  printf("%d", *j);
  return 0;
}
```

a) Compilation error

b) Some Garbage value

c) 0

d) Nothing prints

54. What is the output of this program?

```
#include <stdio.h>
#include  <stdlib.h>
int main() {
  int *numbers = (int*)calloc(4, sizeof(int));
  numbers[0] = 2;
  free(numbers);
  printf("Stored integers are ");
  printf("numbers[%d] = %d ", 0, numbers[0]);
  return 0;
}
```

a) Garbage value

b) 2

c) 0

d) Compilation error

55. What is the output of this program?

```
#include <stdio.h>
void main(){
  int *ptr = (int *)malloc(sizeof(int));
  *ptr = 10;
  free(ptr);
  p = 5;
```

```
   printf("%d", ptr);
}
```

a) Compilation error

b) 5

c) 0

d) Garbage value

56. What is the output of this program?

```
#include <stdio.h>
#include  <stdlib.h>
int main(){
  int *p;
  p = (int *)malloc(40);
  printf("%d", sizeof(p));
  free(p);
  return 0;
}
```

a) 40

b) 50

c) 30

d) 20

57. What is the output of this program?

```
#include <stdio.h>
#include  <stdlib.h>
int main(){
  struct test{
    int i;
    float f;
    char c;
  };
  struct test *ptr;
  ptr = (struct test *)malloc(sizeof(struct test));
```

```
  ptr ->f = 2.5f;
  printf("%f", ptr->f);
  return 0;
}
```

a) Compilation error

b) 2.5

c) Garbage value

d) 0

58. What is another name for the circular queue among the following options?

a) Square buffer

b) Rectangle buffer

c) Ring buffer

d) None of the above

59. In the linked list implementation of queue, where will the new element be inserted?

a) At the middle position of the linked list

b) At the head position of the linked list

c) At the tail position of the linked list

d) None of the above

60. Which of the following option is true if implementation of Queue is from the linked list?

a) In enqueue operation, new nodes are inserted from the beginning and in dequeue operation, nodes are removed from the end.

b) In enqueue operation, new nodes are inserted from the end and in dequeue operation, nodes are deleted from the beginning.

c) In enqueue operation, new nodes are inserted from the end and in dequeue operation, nodes are deleted from the end.

d) Both a and b

61. Consider the implementation of the singly linked list having the head pointer only in the representation. Which of the following operations can be performed in O(1) time?

i) Deletion of the last node in the linked list

ii) Insertion at the front of the linked list

iii) Deletion of the first node in the linked list

iv) Insertion at the end of the linked list

a) ii

b) both ii and iii

c) both i and iv

d) both i and ii

62. What would be the time complexity if user tries to insert the element at the end of the linked list (head pointer is known)?

a) O(1)

b) O(n)

c) O(logn)

d) O(nlogn)

63. Which of the following is the time complexity to search an element in the linked list?

a) O(1)

b) O(n)

c) O(logn)

d) O(nlogn)

64. How do you calculate the pointer difference in a memory efficient double linked list?

a) head xor tail

b) pointer to previous node xor pointer to next node

c) pointer to previous node – pointer to next node

d) pointer to next node – pointer to previous node

Answer: b

65. What is the worst case time complexity of inserting a node in a doubly linked list?

a) O(nlogn)

b) O(logn)

c) O(n)

d) O(1)

66. Which of the following application makes use of a circular linked list?

a) Undo operation in a text editor

b) Recursive function calls

c) Allocating CPU to resources

d) Implement Hash Tables

67. Which of the following is false about a circular linked list?

a) Every node has a successor

b) Time complexity of inserting a new node at the head of the list is O(1)

c) Time complexity for deleting the last node is O(n)

d) We can traverse the whole circular linked list by starting from any point

68. Consider a small circular linked list. How to detect the presence of cycles in this list effectively?

a) Keep one node as head and traverse another temp node till the end to check if its ‘next points to head

b) Have fast and slow pointers with the fast pointer advancing two nodes at a time and slow pointer advancing by one node at a time

c) Cannot determine, you have to pre-define if the list contains cycles

d) Circular linked list itself represents a cycle. So no new cycles cannot be generated

69. Which of the following header files must necessarily be included to use dynamic memory allocation functions?

a) stdlib.h

b) stdio.h

c) memory.h

d) dos.h

70. The type of linked list in which the node does not contain any pointer or reference to the previous node is _____

a) Circularly singly linked list

b) Singly linked list

c) Circular doubly linked list

d) Doubly linked list

71. The advantage of using linked lists over arrays is that _____

a) Linked list is an example of linear data structure

b) Insertion and deletion of an element can be done at any position in a linked list

c) Linked list can be used to store a collection of homogenous and heterogeneous data types

d) The size of a linked list is fixed

72. Which function is used to delete the allocated memory space?

a) Dealloc()

b) free()

c) Both A and B

d) None of the above

73. Which of the following is/are true

a) calloc() allocates the memory and also initializes the allocates memory to zero, while memory allocated using malloc() has random data.

b) malloc() and memset() can be used to get the same effect as calloc()

c) Both malloc() and calloc() return 'void *' pointer

d) All of the above

74. Which of the following is true?

a) "ptr = calloc(m, n)" is equivalent to following

b) r = malloc(m * n);

c) "ptr = calloc(m, n)" is equivalent to following

d) r = malloc(m * n); memset(ptr, 0, m * n);

75. Specify the 2 library functions to dynamically allocate memory?

a) malloc() and memalloc()

b) alloc() and memalloc()

c) malloc() and calloc()

d) memalloc() and faralloc()

76. malloc() returns a float pointer if memory is allocated for storing float's and a double pointer if memory is allocated for storing double's.

a) TRUE

b) FALSE

c) May Be

d) Can't Say

77. Which statment is true about the given code ?

#include <stdio.h>

#include  <stdlib.h>

```
int main(){
   int *a[5];
   a = (int*) malloc(sizeof(int)*5);
   free(a);
   return 0;
}
```

a) Error: unable to allocate memory

b) Error: We cannot store address of allocated memory in a

c) Error: unable to free memory

d) No error

78. What is the Error of this program?

```
#include <stdio.h>
#include  <stdlib.h>
int main() {
    char *ptr;
    *ptr = (char)malloc(30);
    strcpy(ptr, "RAM");
    printf("%s", ptr);
    free(ptr);
    return 0;
}
```

a) Error: in strcpy() statement.

b) Error: in *ptr = (char)malloc(30);

c) Error: in free(ptr);

d) No error

79. How will you free the memory allocated by the following program?

```
#include <stdio.h>
#include  <stdlib.h>
#define MAXROW 2
#define MAXCOL 3
int main() {
```

```
    int **p, i, j;
    p = (int **) malloc(MAXROW * sizeof(int*));
    return 0;
}
```

a) memfree(int p);

b) dealloc(p);

c) malloc(p, 0);

d) free(p);

80. In a circular linked list

a) Components are all linked together in some sequential manner.

b) There is no beginning and no end.

c) Components are arranged hierarchically.

d) Forward and backward traversal within the list is permitted.

81. A linear collection of data elements where the linear node is given by means of pointer is called?

a) Linked list

b) Node list

c) Primitive list

d) None

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

a) Deleting a node whose location in given

b) Searching of an unsorted list for a given item

c) Inverting a node after the node with given location

d) Traversing a list to process each node

83. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head and tail pointer. Given the representation, which of the following operation can be implemented in O(1) time?

I) Insertion at the front of the linked list

II) Insertion at the end of the linked list

III) Deletion of the front node of the linked list

IV) Deletion of the last node of the linked list

a) I and II
b) I and III
c) I,II and III
d) I,II and IV

84. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?

I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the last node of the linked list

a) I and II
b) I and III
c) I,II and III
d) I,II and IV

85. Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer and tail pointer. Given the representation, which of the following operation can be implemented in O(1) time?

I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list

a) I and II
b) I and III
c) I,II and III
d) I,II,III and IV

86. Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list

a) I and II
b) I and III
c) I,II and III
d) I,II,III and IV

87. Consider an implementation of unsorted circular linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list

a) I and II
b) I and III
c) I, II, III and IV
d) None

88. Consider an implementation of unsorted circular doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list

a) I and II

b) I and III

c) I, II and III

d) I,II,III and IV

89. In linked list each node contain minimum of two fields. One field is data field to store the data second field is?

a) Pointer to character

b) Pointer to integer

c) Pointer to node

d) Node

90. What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?

a) O(1)

b) O(n)

c) θ (n)

d) θ (1)

91. What would be the asymptotic time complexity to insert an element at the second position in the linked list?

a) O(1)

b) O(n)

c) O(n2)

d) None

92. The concatenation of two list can performed in O(1) time. Which of the following variation of linked list can be used?

a) Singly linked list

b) Doubly linked list

c) Circular doubly linked list

d) Array implementation of list

93. Consider the following definition in c programming language

struct node{

int data;

struct node * next;

}

typedef struct node NODE;

NODE *ptr;

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

a) ptr=(NODE*)malloc(sizeof(NODE));

b) ptr=(NODE*)malloc(NODE);

c) ptr=(NODE*)malloc(sizeof(NODE*));

d) ptr=(NODE)malloc(sizeof(NODE));

94. A variant of linked list in which last node of the list points to the first node of the list is?

a) Singly linked list

b) Doubly linked list

c) Circular linked list

d) Multiply linked list

95. In doubly linked lists, traversal can be performed?

a) Only in forward direction

b) Only in reverse direction

c) In both directions

d) None

96. What kind of linked list is best to answer question like “What is the item at position n?”

a) Singly linked list

b) Doubly linked list

c) Circular linked list

d) Array implementation of linked list

97. A variation of linked list is circular linked list, in which the last node in the list points to first node of the list. One problem with this type of list is?

a) It waste memory space since the pointer head already points to the first node and thus the list node does not need to point to the first node.

b) It is not possible to add a node at the end of the list.

c) It is difficult to traverse the list as the pointer of the last node is now not NULL

d) All of above

98. A variant of the linked list in which none of the node contains NULL pointer is?

a) Singly linked list

b) Doubly linked list

c) Circular linked list

d) None

99. In circular linked list, insertion of node requires modification of?

a) One pointer

b) Two pointer

c) Three pointer

d) None

100. Which of the following statements about linked list data structure is/are TRUE?

a) Addition and deletion of an item to/ from the linked list require modification of the existing pointers

b) The linked list pointers do not provide an efficient way to search an item in the linked list

c) Linked list pointers always maintain the list in ascending order

d) The linked list data structure provides an efficient way to find kth element in the list

101. Linked lists are not suitable to for the implementation of?

a) Insertion sort

b) Radix sort

c) Polynomial manipulation

d) Binary search

102. In worst case, the number of comparison need to search a singly linked list of length n for a given element is

a) log n

b) n/2

c) log2 n-1

d) n

103. consider the function f defined here:

```
struct item{
int data;
```

```
struct item * next;
};
int f (struct item *p){
return((p==NULL) ||((p->next==NULL)||(p->data<=p->next->data) && (p->next)));
}
```

For a given linked list p, the function f returns 1 if and only if

a) the list is empty or has exactly one element

b) the element in the list are sorted in non-decreasing order of data value

c) the element in the list are sorted in non-increasing order of data value

d) not all element in the list have the same data value

104. The following C function takes a singly 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 left blank.

```
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;
}
```

Choose the correct alternative to replace the blank line

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;

105. The following C Function takes a singly- linked list of integers as a parameter and rearranges the elements of the lists. 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

106. Which of the following statement is not true about the doubly linked list?

a. We can traverse in both the directions.

b. It requires extra space

c. Implementation of doubly linked list is easier than the singly linked list

d. It stores the addresses of the next and the previous node

107. Which of the following is not a disadvantage to the usage of array?

a) Fixed size

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

c) Insertion based on position

d) Accessing elements at specified positions

108. What is the time complexity to count the number of elements in the linked list?

a) O(1)

b) O(n)

c) O(logn)

d) O(n2)

109. What is the functionality of the following code?

```
public void function(Node node) {
        if(size == 0)
                head = node;
        else    {
                Node temp,cur;
                for(cur = head; (temp = cur.getNext())!=null; cur = temp);
                cur.setNext(node);
        }
        size++;
}
```

a) Inserting a node at the beginning of the list

b) Deleting a node at the beginning of the list

c) Inserting a node at the end of the list

d) Deleting a node at the end of the list

110. What is the space complexity for deleting a linked list?

a) O(1)

b) O(n)

c) Either O(1) or O(n)

d) O(logn)

111. Which of these is not an application of a linked list?

a) To implement file systems

b) For separate chaining in hash-tables

c) To implement non-binary trees

d) Random Access of elements

112. Which of the following piece of code has the functionality of counting the number of elements in the list?

a)
```
public int length(Node head) {
        int size = 0;
        Node cur = head;
        while(cur!=null)    {
            size++;
            cur = cur.getNext();
        }
        return size;
}
```

b)
```
public int length(Node head) {
      int size = 0;
        Node cur = head;
        while(cur!=null) {
            cur = cur.getNext();
            size++;
        }
        return size;
}
```

c)
```
public int length(Node head) {
        int size = 0;
        Node cur = head;
        while(cur!=null) {
            size++;
```

```
            cur = cur.getNext();
        }
}
```

d)
```
public int length(Node head) {
        int size = 0;
        Node cur = head;
        while(cur!=null) {
            size++;
            cur = cur.getNext().getNext();
        }
        return size;
}
```

113. How do you insert an element at the beginning of the list?

a)
```
public void insertBegin(Node node) {
        node.setNext(head);
        head = node;
        size++;
}
```

b)
```
public void insertBegin(Node node){
        head = node;
        node.setNext(head);
        size++;
}
```

c)
```
public void insertBegin(Node node) {
        Node temp = head.getNext()
        node.setNext(temp);
        head = node;
        size++;
}
```

d)
```
public void insertBegin(Node node) {
        Node temp = head.getNext()
```

```
        node.setNext(temp);
        node = head;
        size++;
}
```

114. What is the functionality of the following piece of code?

```
public int function(int data) {
        Node temp = head;
        int var = 0;
        while(temp != null)    {
                if(temp.getData() == data)     {
                        return var;
                }
                var = var+1;
                temp = temp.getNext();
        }
        return Integer.MIN_VALUE;
}
```

a) Find and delete a given element in the list

b) Find and return the given element in the list

c) Find and return the position of the given element in the list

d) Find and insert a new element in the list

115. What does the following function do for a given Linked List with first node as head?

```
void fun1(struct node* head) {
   if(head == NULL)
   return;
   fun1(head->next);
   printf("%d  ", head->data);
}
```

a) Prints all nodes of linked lists

b) Prints all nodes of linked list in reverse order

c) Prints alternate nodes of Linked List

d) Prints alternate nodes in reverse order

116. Which of the following points is/are not true about Linked List data structure when it is compared with an array?

a) Arrays have better cache locality that can make them better in terms of performance

b) It is easy to insert and delete elements in Linked List

c) Random access is not allowed in a typical implementation of Linked Lists

d) Access of elements in linked list takes less time than compared to arrays

117. What kind of linked list is best to answer questions like “What is the item at position n?”

a) Singly linked list

b) Doubly linked list

c) Circular linked list

d) Array implementation of linked list

118. Linked list is considered as an example of _____ type of memory allocation.

a) Dynamic

b) Static

c) Compile time

d) Heap

119. In Linked List implementation, a node carries information regarding _____

a) Data

b) Link

c) Data and Link

d) Node

120. Linked list data structure offers considerable saving in _____

a) Computational Time

b) Space Utilization

c) Space Utilization and Computational Time

d) Speed Utilization

121. Which of the following points is/are not true about Linked List data structure when it is compared with an array?

a) Arrays have better cache locality that can make them better in terms of performance

b) It is easy to insert and delete elements in Linked List

c) Random access is not allowed in a typical implementation of Linked Lists

d) Access of elements in linked list takes less time than compared to arrays

122. The following function reverse() is supposed to reverse a singly linked list. There is one line missing at the end of the function.

```
/* Link list node */
struct node {
    int data;
    struct node* next;
};
/* head_ref is a double pointer which points to head (or start) pointer
  of linked list */
static void reverse(struct node** head_ref) {
    struct node* prev   = NULL;
    struct node* current = *head_ref;
    struct node* next;
    while (current != NULL)     {
        next  = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    /*ADD A STATEMENT HERE*/
}
```

What should be added in place of "/*ADD A STATEMENT HERE*/", so that the function correctly reverses a linked list.

a) *head_ref = prev;

b) *head_ref = current;

c) *head_ref = next;

d) *head_ref = NULL;

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

1->2->3->4->5->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

124. The following C function takes a simply-linked list as an 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;

125. 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

126. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is?

a) log2 n

b) n⁄2

c) log2 n – 1

d) n

127. 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

128. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?

a) Delete the first element

b) Insert a new element as a first element

c) Delete the last element of the list

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

129. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is?

a) log2 n

b) n⁄2

c) log2 n – 1

d) n

130. Which one of the following node is considered the top of the stack if the stack is implemented using the linked list?

a. First node

b. Second node

c. Last node

d. None of the above

# Unit-IV Tree

131. Which of the following is an example for non linear data type?

a) Tree

b) Array

c) Linked list

d) Queue

132. The operation of processing each element in the list is known as ……

a) sorting

b) merging

c) inserting

d) traversal

133. Another name for the directed graph is …….

a) Direct graph

b) Digraph

c) Dir-graph

d) Digraph

134. Graph G is …….. if for any pair u, v of nodes in G there is a path from u to v or path from v to u.

a) Literally connected

b) Widely Connected

c) Unliterally connected

d) Literally connected

135. In Binary trees, nodes with no successor are called ……

a) End nodes

b) Terminal nodes

c) Final nodes

d) Last nodes

136. A connected graph T without any cycles is called ……..

a) free graph

b) no cycle graph

c) non-cycle graph

d) circular graph

137. Trees are said ……. if they are similar and have the same contents at corresponding nodes.

a) Duplicate

b) Carbon copy

c) Replica

d) Copies

138. A connected graph T without any cycles is called a ……..

a) A tree graph

b) Free tree

c) A tree d

d) All of the above

139. Every node N in a binary tree T except the root has a unique parent called the …… of N.

a) Antecedents

b) Predecessor

c) Forerunner

d) Precursor

140. In a graph if E=(u,v) means ……

a) u is adjacent to v but v is not adjacent to u

b) e begins at u and ends at v

c) u is processor and v is the successor

d) both b and c

141. Sequential representation of binary tree uses ……..

a) Array with pointers

b) Single linear array

c) Two-dimensional arrays

d) Three-dimensional arrays

142. In a graph, if e=[u,v], Then u and v are called ……..

a) Endpoints of e

b) Adjacent nodes

c) Neighbors

d) All of the above

143. TREE[1]=NULL indicates the tree is ……..

a) Overflow

b) Underflow

c) Empty

d) Full

144. A binary tree whose every node has either zero or two children is called …….

a) complete binary tree

b) binary search tree

c) extended binary tree

d) data structure

145. In a 2-tree, nodes with 0 children are called ……

a) Exterior node

b) Outside node

c) Outer node

d) External node

146. Which indicates pre-order traversal?

a) Left sub-tree, Right sub-tree and root

b) Right sub-tree, Left sub-tree and root

c) Root, Left sub-tree, Right sub-tree

d) Right sub-tree, root, Left sub-tree

147. In extended-binary tree nodes with 2 children are called ……..

a) Interior node

b) Domestic node

c) Internal node

d) Inner node

148. A terminal node in a binary tree is called ……

a) Root

b) Leaf

c) Child

d) Branch

149. A directed graph is ……. if there is a path from each vertex to every other vertex in the digraph.

a) Weakly connected

b) Strongly Connected

c) Tightly Connected

d) Linearly Connected

150. In the …….. traversal we process all of a vertex's descendants before we move to an adjacent vertex.

a) Depth First

b) Breadth First

c) With First

d) Depth Limited

151. State True or False.

i) An undirected graph which contains no cycles is called forest.

ii) A graph is said to be complete if there is an edge between every pair of vertices.

a) True, True

b) False, True

c) False, False

d) True, False

152. A graph is said to be …… if the vertices can be split into two sets V1 and V2 such there are no edges between two vertices of V1 or two vertices of V2.

a) Partite

b) Bipartite

c) Rooted

d) Bisects

153. A graph is a collection of nodes, called ……. And line segments called arcs or …….. that connect pair of nodes.

a) vertices, edges

b) edges, vertices

c) vertices, paths

d) graph node, edges

154. A …….. is a graph that has weights of costs associated with its edges.

a) Network

b) Weighted graph

c) Both A and B

d) None A and B

155. In general, the binary search method needs no more than ……. comparisons.

a) [log2n]-1

b) [logn]+1

c) [log2n]

d) [log2n]+1

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

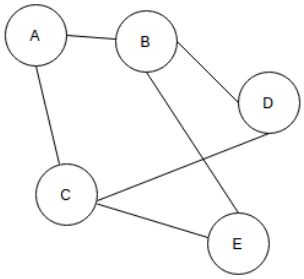
a) Every path is a trail

b) Every trail is a path

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

d) Path and trail have no relation

157. 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

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

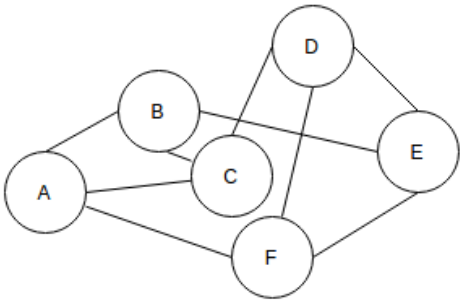
a) (n*(n+1))/2

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

c) n

d) Information given is insufficient

159. The given Graph is regular.




a) True

b) False

160. Which of the following properties does a simple graph not hold?

a) Must be connected

b) Must be unweighted

c) Must have no loops or multiple edges

d) Must have no multiple edges

161. Which of the following is true?

a) A graph may contain no edges and many vertices

b) A graph may contain many edges and no vertices

c) A graph may contain no edges and no vertices

d) A graph may contain no vertices and many edges

162. A graph with all vertices having equal degree is known as a _____

a) Multi Graph

b) Regular Graph

c) Simple Graph

d) Complete Graph

163. Which of the following ways can be used to represent a graph?

a) Adjacency List and Adjacency Matrix

b) Incidence Matrix

c) Adjacency List, Adjacency Matrix as well as Incidence Matrix

d) No way to represent

164. What is the maximum number of children that a binary tree node can have?

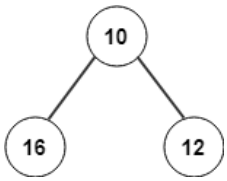
a) 0

b) 1

c) 2

d) 3

165. The following given tree is an example for?




a) Binary tree

b) Binary search tree

c) Fibonacci tree

d) AVL tree

166. How many common operations are performed in a binary tree?

a) 1

b) 2

c) 3

d) 4

167. What is the traversal strategy used in the binary tree?

a) depth-first traversal

b) breadth-first traversal

c) random traversal

d) Priority traversal

168. How many types of insertion are performed in a binary tree?

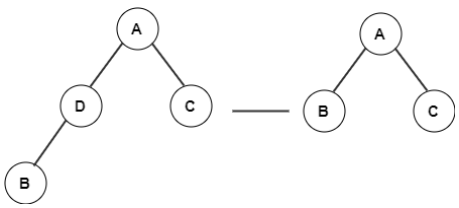
a) 1

b) 2

c) 3

d) 4

169. What operation does the following diagram depict?




a) inserting a leaf node

b) inserting an internal node

c) deleting a node with 0 or 1 child

d) deleting a node with 2 children

170. Which of the following properties are obeyed by all three tree – traversals?

a) Left subtrees are visited before right subtrees

b) Right subtrees are visited before left subtrees

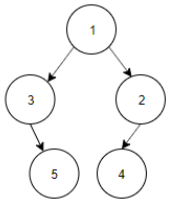
c) Root node is visited before left subtree

d) Root node is visited before right subtree

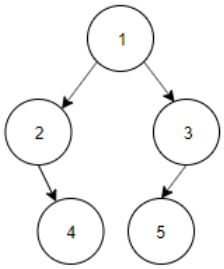
171. Construct a binary tree using the following data.

The preorder traversal of a binary tree is 1, 2, 5, 3, 4. The inorder traversal of the same binary tree is 2, 5, 1, 4, 3.

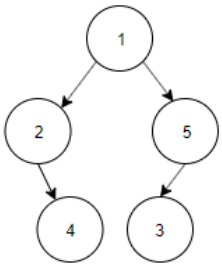
a)



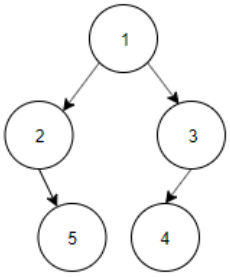

b)



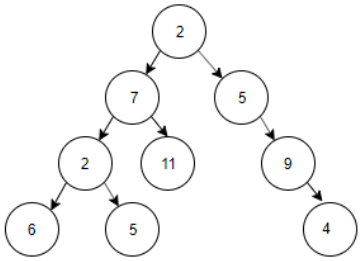

c)



d)




172. For the tree below, write the pre-order traversal.



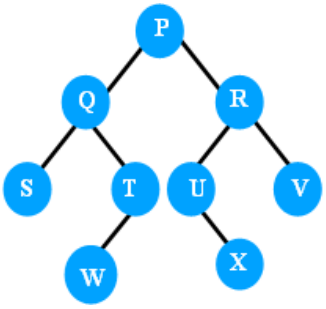

a) 2, 7, 2, 6, 5, 11, 5, 9, 4

b) 2, 7, 5, 2, 6, 9, 5, 11, 4

c) 2, 5, 11, 6, 7, 4, 9, 5, 2

d) 2, 7, 5, 6, 11, 2, 5, 4, 9

173. Find the postorder traversal of the binary tree shown below.



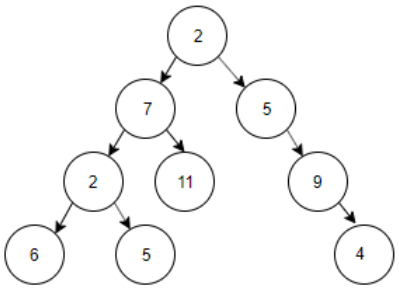

a) P Q R S T U V W X

b) W R S Q P V T U X

c) S W T Q X U V R P

d) S T W U X V Q R P

174. For the tree below, write the in-order traversal.



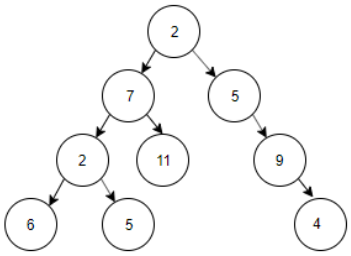

a) 6, 2, 5, 7, 11, 2, 5, 9, 4

b) 6, 5, 2, 11, 7, 4, 9, 5, 2

c) 2, 7, 2, 6, 5, 11, 5, 9, 4

d) 2, 7, 6, 5, 11, 2, 9, 5, 4

175. For the tree below, write the preorder traversal.




a) 2, 7, 2, 6, 5, 11, 5, 9, 4

b) 2, 7, 2, 6, 5, 11, 5, 9, 4

c) 2, 5, 11, 6, 7, 4, 9, 5, 2

d) 2, 7, 5, 6, 11, 2, 5, 4, 9

## Unit-V Graph

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

a) Every path is a trail

b) Every trail is a path

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

d) Path and trail have no relation

177. In the given graph identify the cut vertices.

a) B and E

b) C and D

c) A and E

d) C and B

178. 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

179. 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

180. The given Graph is regular.

a) True

b) False

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

a) True

b) False

182. A connected planar graph having 6 vertices, 7 edges contains _____ regions.

a) 15

b) 3

c) 1

d) 11

183. If a simple graph G, contains n vertices and m edges, the number of edges in the Graph G'(Complement of G) is _____

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

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

c) (n*n-n-2*m)/2

d) (n*n-n+2*m)/2

184. Which of the following properties does a simple graph not hold?

a) Must be connected

b) Must be unweighted

c) Must have no loops or multiple edges

d) Must have no multiple edges

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

a) 24

b) 21

c) 25

d) 16

186. Which of the following is true?

a) A graph may contain no edges and many vertices

b) A graph may contain many edges and no vertices

c) A graph may contain no edges and no vertices

d) A graph may contain no vertices and many edges

187. For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true?

a) v=e

b) v = e+1

c) v + 1 = e

d) v = e-1

188. 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

189. A graph with all vertices having equal degree is known as a _____

a) Multi Graph

b) Regular Graph

c) Simple Graph

d) Complete Graph

190. Which of the following ways can be used to represent a graph?

a) Adjacency List and Adjacency Matrix

b) Incidence Matrix

c) Adjacency List, Adjacency Matrix as well as Incidence Matrix

d) No way to represent

## Unit-VI Hashing and Files

191. What is a hash table?

a) A structure that maps values to keys

b) A structure that maps keys to values

c) A structure used for storage

d) A structure used to implement stack and queue

192. If several elements are competing for the same bucket in the hash table, what is it called?

a) Diffusion

b) Replication

c) Collision

d) Duplication

193. What is direct addressing?

a) Distinct array position for every possible key

b) Fewer array positions than keys

c) Fewer keys than array positions

d) Same array position for all keys

194. What is the search complexity in direct addressing?

a) O(n)

b) O(logn)

c) O(nlogn)

d) O(1)

195. What is a hash function?

a) A function has allocated memory to keys

b) A function that computes the location of the key in the array

c) A function that creates an array

d) A function that computes the location of the values in the array

196. Which of the following is not a technique to avoid a collision?

a) Make the hash function appear random

b) Use the chaining method

c) Use uniform hashing

d) Increasing hash table size

197. What is the load factor?

a) Average array size

b) Average key size

c) Average chain length

d) Average hash table length

198. What is simple uniform hashing?

a) Every element has equal probability of hashing into any of the slots

b) A weighted probabilistic method is used to hash elements into the slots

c) Elements has Random probability of hashing into array slots

d) Elements are hashed based on priority

199. In simple uniform hashing, what is the search complexity?

a) O(n)

b) O(logn)

c) O(nlogn)

d) O(1)

200. In simple chaining, what data structure is appropriate?

a) Singly linked list

b) Doubly linked list

c) Circular linked list

d) Binary trees