

Attempt ALL the following questionsChoose the **MOST APPROPRIATE** answer for the following statements.

You may choose E (=ALL) if all answers (A, B, C and D) are correct or choose F (=NONE) if none of the answers fits.

Please write your answers on the ANSWER SHEET ONLY

In the designated answer sheet, mark your choice (a, b, c, d, e, or f) in front of the question number.

Be sure that you have filled the appropriate bubbles carefully as in the example below.Example: if the choice for question 300 is "C" then your answer sheet should look like this:300. (a) (b) ☒ (c) (d) (e) (f)

- Which of these statements best describes an array?
 - A data structure that shows a hierarchical behaviour
 - Container of objects of similar types
 - Arrays are immutable once initialised
 - Array is not a data structure
- How do you instantiate an array in Java?
 - int arr[] = new int(3);
 - int arr[];
 - int arr[] = new int[3];
 - int arr() = new int(3);
- Which of the following is the correct way to declare a multidimensional array in Java?
 - int[] arr;
 - int arr[][];
 - int[][] arr;
 - int [[]] arr;
- What is the output of the following Java code?


```
public class array
{
    public static void main (String args[])
    {int []arr = {1,2,3,4,5}; System.out.println(arr[2]); System.out.println(arr[4]);}
```

 - 3 and 5
 - 5 and 3
 - 2 and 4
 - 4 and 2
- What is the output of the following Java code?


```
public class array
{
    public static void main (String args[])
    {int []arr = {1,2,3,4,5}; System.out.println(arr[5]);}
```

 - 4
 - 5
 - ArrayIndexOutOfBoundsException
 - InavlidInputException
- When does the ArrayIndexOutOfBoundsException occur?
 - Compile-time
 - Run-time
 - Not an error
 - Not an exception at all
- What are the advantages of arrays?
 - Objects of mixed data types can be stored
 - Elements in an array cannot be sorted
 - Index of first element of an array is 1
 - Easier to store elements of same data type
- What are the disadvantages of arrays?
 - Data structure like queue or stack cannot be implemented
 - There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
 - Index value of an array can be negative
 - Elements are sequentially accessed
- Assuming int is of 4bytes, what is the size of int arr[15];?
 - 15
 - 19
 - 11
 - 60
- Elements in an array are accessed
 - randomly
 - sequentially
 - exponentially
 - logarithmically
- Process of inserting an element in stack is called
 - Create
 - Push
 - Evaluation
 - Pop
- Process of removing an element from stack is called

- a) Create b) Push c) Evaluation d) Pop
13. In a stack, if a user tries to remove an element from an empty stack it is called
- a) Underflow b) Empty collection c) Overflow d) Garbage Collection
14. Entries in a stack are "ordered". What is the meaning of this statement?
- a) A collection of stacks is sortable c) The entries are stored in a linked list
- b) Stack entries may be compared with the '<' operation d) There is a Sequential entry that is one by one
15. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: $((()())())$?
- a) 1 b) 2 c) 3 d) 4 or more
16. A linear collection of data elements where the linear node is given by means of pointer is called?
- a) Linked list b) Node list c) Primitive list d) Unordered list
17. 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 iii) Deletion of the front node of the linked list
- ii) Insertion at the end 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
18. In linked list each node contains a 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
19. What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?
- a) $O(1)$ b) $O(n)$ c) $\theta(n)$ d) $\theta(1)$
20. What would be the asymptotic time complexity to insert an element at the front of the linked list (head is known)?
- a) $O(1)$ b) $O(n)$ c) $O(n^2)$ d) $O(n^3)$
21. What would be the asymptotic time complexity to find an element in the linked list?
- a) $O(1)$ b) $O(n)$ c) $O(n^2)$ d) $O(n^4)$
22. What would be the asymptotic time complexity to insert an element at the second position in the linked list?
- a) $O(1)$ b) $O(n)$ c) $O(n^2)$ d) $O(n^3)$
23. The concatenation of two lists can be performed in $O(1)$ time. Which of the following variation of the linked list can be used?
- a) Singly linked list b) Doubly linked list c) Circular doubly linked list d) Array implementation of list
24. 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
25. Linked lists are not suitable for the implementation of
- a) Insertion sort b) Radix sort c) Polynomial manipulation d) Binary search
26. Linked list is considered as an example of type of memory allocation.
- a) Dynamic b) Static c) Compile time d) Heap
27. Linked list data structure offers considerable saving in
- a) Computational Time b) Space Utilization c) Space Utilization and Computational Time d) Speed Utilization
28. 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
29. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

- a) Insertion Sort b) Quick Sort c) Heap Sort d) Merge Sort
30. 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
31. 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 c) Possible if size of linked list is odd
b) Possible if size of linked list is even d) Possible if X is not first node
32. 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 c) Delete the last element of the list
b) Insert a new element as a first element d) Add a new element at the end of the list
33. 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
34. 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 c) Inserting a node at the end of the list
b) Deleting a node at the beginning of the list d) Deleting a node at the end of the list
35. 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++;}`
36. 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 c) Find and return the position of the given element in the list
b) Find and return the given element in the list d) Find and insert a new element in the list
37. Which of the following is false about a doubly linked list?
a) We can navigate in both the directions c) The insertion and deletion of a node take a bit longer
b) It requires more space than a singly linked list d) Implementing a doubly linked list is easier than singly linked list
38. What is a memory efficient double linked list?
a) Each node has only one pointer to traverse the list back and forth
b) The list has breakpoints for faster traversal
c) An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list
d) A doubly linked list that uses bitwise AND operator for storing addresses
39. What differentiates a circular linked list from a normal linked list?
a) You cannot have the 'next' pointer point to null in a circular linked list
b) It is faster to traverse the circular linked list
c) You may or may not have the 'next' pointer point to null in a circular linked list

- d) Head node is known in circular linked list
40. Which of the following is false about a circular linked list?
- Every node has a successor
 - Time complexity of inserting a new node at the head of the list is $O(1)$
 - Time complexity for deleting the last node is $O(n)$
 - We can traverse the whole circular linked list by starting from any point
41. What is the maximum number of swaps that can be performed in the Selection Sort algorithm?
- $n-1$
 - n
 - 1
 - $n-2$
42. Which of the following is a Divide and Conquer algorithm?
- Bubble sort
 - Selection sort
 - Heap sort
 - Merge sort
43. What will be the best sorting algorithm, given that the array elements are small ($\leq 1e6$)?
- Bubble sort
 - Insertion sort
 - Linear search
 - Selection sort
44. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 2 5 1 7 9 12 11 10 Which statement is correct?
- The pivot could be either the 7 or the 9.
 - The pivot could be the 7, but it is not the 9
 - The pivot is not the 7, but it could be the 9
 - Neither the 7 nor the 9 is the pivot.
45. What is the best time complexity of bubble sort?
- n^2
 - $n \log n$
 - n
 - $n (\log n)^2$
46. Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced)?
- Quick Sort
 - Heap Sort
 - Merge Sort
 - Insertion Sort
47. You have to sort 1 GB of data with only 100 MB of available main memory. Which sorting technique will be most appropriate?
- Heap sort
 - Merge sort
 - Quick sort
 - Insertion sort
48. Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced)?
- Quick Sort
 - Heap Sort
 - Merge Sort
 - Insertion Sort
49. You have an array of n elements. Suppose you implement quicksort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is
- $O(n^2)$
 - $O(n \log n)$
 - $\Theta(n \log n)$
 - $O(n^3)$
50. Which of the following changes to typical Quick Sort improves its performance on average and are generally done in practice?
- Randomly picking up to make worst case less likely to occur.
 - Calling insertion sort for small sized arrays to reduce recursive calls.
 - Quick Sort is tail recursive, so tail call optimizations can be done.
 - A linear time median searching algorithm is used to pick the median, so that the worst-case time reduces to $O(n \log n)$
- 1 and 2
 - 2, 3, and 4
 - 1, 2 and 3
 - 2, 3 and 4
51. Assume that the algorithms considered here sort the input sequences in ascending order. If the input is already in ascending order, which of the following are TRUE ?
- Quicksort runs in $\Theta(n^2)$ time
 - Bubble sort runs in $\Theta(n^2)$ time
 - Merge sort runs in $\Theta(n)$ time
 - Insertion sort runs in $\Theta(n)$ time
- I and II only
 - I and III only
 - II and IV only
 - I and IV only
52. Assume that we use Bubble Sort to sort n distinct elements in ascending order. When does the best case of Bubble Sort occur?
- When elements are sorted in ascending order
 - When elements are not sorted by any order

- b. When elements are sorted in descending order d. There is no best case for Bubble Sort. It always takes $O(n*n)$ time
53. Consider an array of elements $arr[5] = \{5,4,3,2,1\}$, what are the steps of insertions done while doing insertion sort in the array.
- a. 4 5 3 2 1, 3 4 5 2 1, 2 3 4 5 1, 1 2 3 4 5 c. 4 3 2 1 5, 3 2 1 5 4, 2 1 5 4 3, 1 5 4 3 2
- b. 5 4 3 1 2, 5 4 1 2 3, 5 1 2 3 4, 1 2 3 4 5 d. 4 5 3 2 1, 2 3 4 5 1, 3 4 5 2 1, 1 2 3 4 5
54. Which is the correct order of the following algorithms with respect to their time Complexity in the best case?
- a. Merge sort > Quick sort > Insertion sort > selection sort
- b. insertion sort < Quick sort < Merge sort < selection sort
- c. Merge sort > selection sort > quick sort > insertion sort
- d. Merge sort > Quick sort > selection sort > insertion sort
55. A sorting technique is called stable if
- a. It takes $O(n \log n)$ time
- b. It maintains the relative order of occurrence of non-distinct elements
- c. It uses divide and conquer paradigm
- d. It takes $O(n)$ space
56. Quicksort is run on two inputs shown below to sort in ascending order taking first element as pivot,
- (i) 1, 2, 3,, n
- (ii) n, n-1, n-2,, 2, 1
- Let C1 and C2 be the number of comparisons made for the inputs (i) and (ii) respectively. Then,
- a. $C1 < C2$ b. $C1 > C2$ c. $C1 = C2$ d. We cannot say anything for arbitrary n
57. Which one of the following in place sorting algorithms needs the minimum number of swaps?
- a. Quick sort b. Insertion sort c. Selection sort d. Heap sort
58. Selection sort algorithm design technique is an example of
- a. Greedy method b. Divide-and-conquer c. Dynamic Programming d. Backtracking
59. If the address of a pointer of type double is 0xFF0040, if apply the arithmetic operation +5, what will be the new value of the pointer?
- a. 0xFF0048 b. 0xFF0088 c. 0xFF0080 d. 0xFF00F8
60. What is the worst-case time for binary search finding a single item in an array?
- a. Constant time b. Logarithmic time c. Linear time d. Quadratic time
61. What does a run-time complexity usually count?
- a. The number of arithmetic and other operations required for the program to run
- b. The number of megabytes required for the program to run
- c. The number of seconds required for the program to run
- d. The number of seconds plus the number of megabytes
- e. The number of seconds times the number of arithmetic operations
62. In linked list each node contains 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.

Best Wishes