1. In ADT which is the abstracted concept  
   A) What a Data-type can do.  
   B) How it is done.  
   C) Both A & B.  
   D) Nor A neither B. 

2. Order of growth specifies  
   A) How large is our algorithm  
   B) Change in behaviour of an algorithm according to the input size  
   C) Change in the algorithm size as the algorithm changes  
   D) None of the above 

3. LIFO is....  
   A) Last In Fist Out  
   B) Last In Fast out  
   C) Least In First Out  
   D) None of the above 

4. Two main measures for the efficiency of an algorithm are  
   A) Processor and memory  
   B) Complexity and capacity  
   C) Time and space  
   D) Data and space 

5. Which of the following algorithm design technique is used in the quick sort algorithm?  
   A) Dynamic programming  
   B) Backtracking  
   C) Divide and conquer  
   D) Greedy method
6. The running time of an algorithm \( T(n) \), where \( 'n' \) is the input size, is given by
\[
T(n) = 8T(n/2) + qa, \text{ if } n>1
\]
Where \( p, q \) are constants. The order of this algorithm is
A) \( n^2 \)  
B) \( n^n \)  
C) \( n^3 \)  
D) \( n \)  

7. How to describe an algorithm
\( * \) Natural Language like English.\( * \) Flowcharts.\( * \) Pseudo-code.  
A) Only a.  
B) Only b & c.  
C) All a, b, c.  
D) None  

8. When new data is to be inserted into a data structure, but there is no available space; this situation is usually called 
A) underflow  
B) overflow  
C) housefull  
D) saturated  

9. Which of the following are the levels of implementation of data structure
A) Abstract Level  
B) Implementation level  
C) Application level  
D) All of the above  

10. The average successful search time for sequential search on \( 'n' \) items is 
A) \( n/2 \)  
B) \( (n-1)/2 \)  
C) \( (n+1)/2 \)  
D) \( \log (n)+1 \)  

11. Let \( P \) be a quick sort program to sort numbers in ascending order using the first element as pivot. Let \( t1 \) and \( t2 \) be the number of comparisons made by \( P \) for the inputs \{1, 2, 3, 4, 5\} and \{4, 1, 5, 3, 2\} respectively. Which one of the following holds?
A) \( t1 = 5 \)  
B) \( t1 < t2 \)  
C) \( t1 > t2 \)  
D) \( t1 = t2 \)
12. Consider the following C program that attempts to locate an element x in an array Y[]. Using binary search. The program is erroneous.

```c
1. f(int Y[10], int x) {
2.   int i, j, k;
3.   i = 0; j = 9;
4.   do {
5.     k = (i + j) / 2;
6.     if (Y[k] < x) i = k; else j = k;
7.   } while(Y[k] != x && i < j);
8.   if(Y[k] == x) printf("x is in the array ");
9.   else printf("x is not in the array ");
10. }
On which

A)  Y is [1 2 3 4 5 6 7 8 9 10] and x < 10  B)  Y is [1 3 5 7 9 11 13 15 17 19] and x < 1
C)  Y is [2 2 2 2 2 2 2 2 2] and x > 2  D)  Y is [2 4 6 8 10 12 14 16 18 20] and 2 < x < 20 and x is even

13. Consider the following graph:

Which one of the following cannot be the sequence of edges added, in that order, to a minimum spanning tree using
Kruskal's algorithm?

A)  (a—b), (d—f), (b—f), (d—c), (d—e)  B)  (a—b), (d—f), (d—c), (b—f), (d—e)
C)  (d—f), (a—b), (d—c), (b—f), (d—e)  D)  (d—f), (a—b), (d—f), (d—e), (d—c)

14. Which one of the following correctly determines the solution of the recurrence relation with
T(1) = 1? T(n) = 2T(n/2) + Log n

A)  Θ(n)  B)  Θ(n Log n)
C)  Θ(n^2)  D)  Θ(log n)

15. Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?
f1(n) = 2^n f2(n) = n^(3/2) f3(n) = n Log n f4(n) = n^(Log n)

A)  f3, f2, f4, f1  B)  f3, f2, f1, f4
C)  f2, f3, f1, f4  D)  f2, f3, f4, f1
16. Let $w(n)$ and $A(n)$ denote respectively, the worst case and average case running time of an algorithm executed on an input of size $n$. Which of the following is ALWAYS TRUE? (2)

A) $A(n) = \Omega(W(n))$  
B) $A(n) = \Theta(W(n))$  
C) $A(n) = O(W(n))$  
D) $A(n) = o(W(n))$

17. The time complexity of the following C function is (assume $n > 0$)

```c
int recursive (mt n)
{
    if (n == 1)
        return (1);
    else
        return (recursive (n-1) + recursive (n-1));
}
```

A) $O(n)$  
B) $O(n \log n)$  
C) $O(n^2)$  
D) $O(2^n)$

18. Frequency count for the following code is:

```c
Sum_Array( arr[], n)
Step 1. i = 0;
Step 2. s = 0;
Step 3. while i < n
Step 4. s = s + arr[i]
Step 5. i = i + 1;
Step 6. end while;
Step 7. return s;
```

A) $3n+5$  
B) $3n+4$  
C) $3n$  
D) $n$

19. Let $w$ be the minimum weight among all edge weights in an undirected connected graph. Let $e$ be a specific edge of weight $w$. Which of the following is FALSE? (2)

A) There is a minimum spanning tree containing $e$.  
B) If $e$ is not in a minimum spanning tree $T$, then in the cycle formed by adding $e$ to $T$, all edges have the same weight.  
C) Every minimum spanning tree has an edge of weight $w$.  
D) $e$ is present in every minimum spanning tree.
20. In the following C function, let \( n \geq m \).

```c
int gcd(n, m)
{
    if (n % m == 0) return m;
    n = n % m;
    return gcd(n, m);
}
```
How many recursive calls are made by this function?

A) \( \Theta(\log n) \)  
B) \( \Omega(n) \)  
C) \( \Theta(n \log n) \)  
D) \( \Theta(\sqrt{n}) \) 

21. Time complexity for the following code is:

```c
Mat_add(a[], n, b[])  
Step 1. i = 0; j = 0  
Step 2. while i < n  
    while j < n+1  
    Step 3. if i < n+1  
    Step 4. while j < n  
    Step 5. c[i][j] = a[i][j] + b[i][j]  
    Step 6. j = j + 1  
    Step 7. end inner while  
    Step 8. i = i + 1  
    Step 9. end outer while  
    Step 10. return c;
```

A) \( O(n) \)  
B) \( O(n^2) \)  
C) \( O(\log n) \)  
D) \( O(n \log n) \) 

22. Consider the following functions:
\( f(n) = 2^n \)  
\( g(n) = n! \)  
\( h(n) = n^\log n \)
Which of the following statements about the asymptotic behaviour of \( f(n) \), \( g(n) \), and \( h(n) \) is true?

A) \( f(n) = O(g(n)); g(n) = O(h(n)) \)  
B) \( f(n) = \Omega(g(n)); g(n) = O(h(n)) \)  
C) \( g(n) = O(f(n)); h(n) = O(f(n)) \)  
D) \( h(n) = O(f(n)); g(n) = \Omega(f(n)) \) 

23. Which of the following statements is TRUE?

A) The algorithm uses dynamic programming paradigm  
B) The algorithm has a linear complexity and uses branch and bound paradigm  
C) The algorithm has a non-linear polynomial complexity and uses branch and bound paradigm  
D) The algorithm uses divide and conquer paradigm.
24. Suppose $T(n) = 2T(n/2) + n$, $T(0) = T(1) = 1$ Which one of the following is false. 

<table>
<thead>
<tr>
<th>Option</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A)</td>
<td>$T(n) = O(n^2)$</td>
</tr>
<tr>
<td>B)</td>
<td>$T(n) = \Theta(n \log n)$</td>
</tr>
<tr>
<td>C)</td>
<td>$T(n) = \Omega(n^2)$</td>
</tr>
<tr>
<td>D)</td>
<td>$T(n) = O(n \log n)$</td>
</tr>
</tbody>
</table>

25. What is the number of swaps required to sort $n$ elements using selection sort, in the worst case?

<table>
<thead>
<tr>
<th>Option</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A)</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td>B)</td>
<td>$\Theta(n \log n)$</td>
</tr>
<tr>
<td>C)</td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>D)</td>
<td>$\Theta(n n^2 \log n)$</td>
</tr>
</tbody>
</table>

26. Two-way list may be maintained in memory by means of ............

<table>
<thead>
<tr>
<th>Option</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>A)</td>
<td>Queues</td>
</tr>
<tr>
<td>B)</td>
<td>Linear arrays</td>
</tr>
<tr>
<td>C)</td>
<td>Non linear arrays</td>
</tr>
<tr>
<td>D)</td>
<td>Stacks</td>
</tr>
</tbody>
</table>

27. If the address of $A[1][1]$ and $A[2][1]$ are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in ......... order.

<table>
<thead>
<tr>
<th>Option</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>A)</td>
<td>row major</td>
</tr>
<tr>
<td>B)</td>
<td>column major</td>
</tr>
<tr>
<td>C)</td>
<td>matrix major</td>
</tr>
<tr>
<td>D)</td>
<td>none of these</td>
</tr>
</tbody>
</table>

28. Which of the following statements are correct about 6 used in the program?

```c
int num[6];
num[6]=21;
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>A)</td>
<td>In the first statement 6 specifies a particular element, whereas in the second statement it specifies a type.</td>
</tr>
<tr>
<td>B)</td>
<td>In the first statement 6 specifies a array size, whereas in the second statement it specifies a particular element of array.</td>
</tr>
<tr>
<td>C)</td>
<td>In the first statement 6 specifies a particular element, whereas in the second statement it specifies a array size</td>
</tr>
<tr>
<td>D)</td>
<td>In both the statement 6 specifies array size.</td>
</tr>
</tbody>
</table>

29. What is a array?

<table>
<thead>
<tr>
<th>Option</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A)</td>
<td>An array is a series of elements of the same type in contiguous memory locations</td>
</tr>
<tr>
<td>B)</td>
<td>An array is a series of element</td>
</tr>
<tr>
<td>C)</td>
<td>An array is a series of elements of the same type placed in non-contiguous memory locations</td>
</tr>
<tr>
<td>D)</td>
<td>None of the mentioned</td>
</tr>
</tbody>
</table>
30. In C, if you pass an array as an argument to a function, what actually gets passed

A) Value of elements in array  B) First element of the array
C) Base address of the array    D) Address of the last element of array

31. Which of the following gives the memory address of the first element in array?

A) array[0];  B) array[1];
C) array(2);    D) array;

32. Binary search algorithm cannot be applied to

A) sorted linked list  B) sorted binary trees
C) sorted linear array  D) pointer array

33. Which of the following is correct way to define the function `fun()` in the below program?

```c
#include<stdio.h>
int main()
{
    int a[3][4];
    fun(a);
    return 0;
}

A) void fun(int p[][4])
   {
   }
B) void fun(int *p[4])
   {
   }
C) void fun(int *p[][4])
   {
   }
D) void fun(int *p[3][4])
   {
   }
```

34. Each array declaration need not give, implicitly or explicitly, the information about

A) the name of array  B) the data type of array
C) the first data from the set to be stored  D) the index set of the array

35. To implement Sparse matrix dynamically, the following data structure is used

A) Trees  B) Graphs
C) Priority Queues  D) Linked List
36. What will be the output of the program?

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

A) 1, 1, 1 1, 2, 3, 2, 3 3, 2, 3, 2 4, 4, 4, 4  
B) 1, 2, 1, 2 2, 3, 2, 3 3, 4, 3, 4 4, 2, 4, 2  
C) 1, 1, 1 2, 2, 2, 2 2, 2, 2, 2 3, 3, 3, 3  
D) 1, 2, 3, 4 2, 3, 4, 1 3, 4, 1, 2 4, 1, 2, 3

37. What is the output of this program?

```c
#include <stdio.h>
using namespace std;
int main()
{
    int array[3] = {10, 20, 30};
    cout << -2[array];
    return 0;
}
```

A) -15  
B) -30  
C) Compile time error  
D) Garbage Value

38. An element in an array X is called a leader if it is greater than all elements to the right of it in X. The best algorithm to find all leaders in an array

A) Solves it in linear time using a left to right pass of the array  
B) Solves it in linear time using a right to left pass of the array  
C) Solves it using divide and conquer in time 8(nlogn)  
D) Solves it in time 8(n2)
39. What will be the output of this program?

```c
#include <stdio.h>
using namespace std;

int main ()
{
    int array[] = {0, 2, 4, 6, 7, 5, 3};
    int n, result = 0;
    for (n = 0; n < 8; n++) {
        result += array[n];
    }
    cout << result;
    return 0;
}
```

A) 25  B) 26  C) 27  D) None

40. Four matrices M1, M2, M3 and M4 of dimensions pxq, qxr, rxs and sxt respectively can be multiplied in several ways with different number of total scalar multiplications. For example, when multiplied as ((M1 X M2) X (M3 X M4)), the total number of multiplications is pqr + rst + prt. When multiplied as (((M1 X M2) X M3) X M4), the total number of scalar multiplications is pqr + prs + pst. If p = 10, q = 100, r = 20, s = 5 and t = 80, then the number of scalar multiplications needed is

A) 248000  B) 44000  
C) 19000  D) 25000

41. What will be the output of the program?

```c
#include<stdio.h>

int main()
{
    int a[5] = {51, 1, 5, 20, 25};
    int x, y, z;
    x = ++a[1];
    y = a[1]++;
    z = a[x++];
    printf("%d, %d, %d", x, y, z);
    return 0;
}
```

A) 2, 3, 20  B) 2, 1, 5
42. A program P reads in 500 integers in the range [0..100] representing the scores of 500 students. It then prints the frequency of each score above 50. What would be the best way for P to store the frequencies?

A) An array of 50 numbers  
B) An array of 100 numbers  
C) An array of 500 numbers  
D) A dynamically allocated array of 550 numbers

43. If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue?

A) rear=(rear%1)+MAX_SIZE  
B) rear=rear%(MAX_SIZE+1)  
C) rear=(rear+1)%MAX_SIZE  
D) rear=rear+(1%MAX_SIZE)

44. What will be the output of the program?

```c
#include<stdio.h>
int main()
{
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = a[1];
    j = a[1]++;
    m = a[i++];
    printf("%d, %d, %d", i, j, m);
    return 0;
}
```

A) 2, 1, 15  
B) 1, 2, 5  
C) 3, 2, 15  
D) 2, 3, 20

45. What will be the output of the program?

```c
#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;
    printf("%d, %d, %d\n", *ptr, ++ptr, **ptr);
    *p++;
    printf("%d, %d, %d\n", *ptr, ++ptr, **ptr);
    *p++;
    printf("%d, %d, %d\n", *ptr, ++ptr, **ptr);
    return 0;
}
```

A) 0, 0, 0 1, 1 2, 2 3, 3, 3  
B) 1, 1, 2 2, 2 3, 3, 3, 4 4, 4, 1  
C) 1, 1, 1 2, 2 3, 3, 3, 4, 4  
D) 0, 1, 2 1, 2, 3 2, 3, 4 3, 4, 5
46. What will be the output of the program?
```c
#include<stdio.h>
void fun(int **p);
int main()
{
    int a[3][4] = {{1, 2, 3, 4, 4, 3, 2, 8, 7, 8, 9, 0}};
    int *ptr;
    ptr = &a[0][0];
    fun(ptr);
    return 0;
}
void fun(int **p)
{
    printf("%d\n", **p);
}
```

A) 1  
B) 2  
C) 3  
D) 4

47. What will be the output of the program?
```c
#include<stdio.h>
int main()
{
    float arr[] = {12.4, 2.3, 4.5, 6.7};
    printf("%d\n", sizeof(arr)/sizeof(arr[0]));
    return 0;
}
```

A) 4  
B) 5  
C) 6  
D) 7

48. What will be the output of the program?
```c
#include<stdio.h>
void main()
{
    float arr[] = {12.4, 2.3, 4.5, 6.7};
    printf("%d", sizeof(arr)/sizeof(arr[0]));
}
```

A) 5  
B) 4  
C) 6  
D) 7

49. What will be the output of the following code?
```c
#include<stdio.h>
void main()
{
    int a[10];
    printf("%d %d", a[-1], a[12]);
}
```

A) 0 0  
B) 0 Garbage Value  
C) Garbage Value 0  
D) Garbage Value Garbage Value
50. Let $A$ be a square matrix of size $n \times n$. Consider the following program. What is the expected output?

```plaintext
C = 100
for i = 1 to n do
    for j = 1 to n do
        Temp = A[i][j] + C
        A[i][j] = A[j][i]
        A[j][i] = Temp - C
    
for i = 1 to n do
    for j = 1 to n do
        Output(A[i][j]);
```

A) The matrix $A$ itself     B) Transpose of matrix $A$
C) Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of $A$     D) None of the above