1.) $\left[\frac{n}{k}\right] = \left[\frac{n-k-1}{k}\right]$ let $n=1$ & $k=1$

$\therefore \left[\frac{1}{1}\right] = \left[\frac{2-1}{1}\right]$

$1 = 1$.

let no. of asume that the statement holds for $n=m$ and $k=l$.

so, $\left[\frac{m}{l}\right] = \left[\frac{m+l-1}{l}\right]$ is True

let $n = m+1$ and $k = l+1$

$\left[\frac{m+1}{l+1}\right] = \left[\frac{m+1+l+1-1}{l+1}\right]$

where $n = m+1$ and $k = l+1$

2.) $\sum_{i=0}^{n} 3^i = \frac{3^{n-1}-1}{2}$ let $n=1$

$\therefore 3^0 + 3^1 = 3$

$4 = 4$

let $n = k$

let assume that the statement hold for $n = k$.

$\therefore \sum^{k} 3^i = \frac{3^{k+1}-1}{2}$ is true

let $n = k+1$

so, $\sum_{i=0}^{k} 3^i = \frac{3^{k+1+1}-1}{2}$

$1 + 3^1 + 3^2 + \dots 3^{k+1} = \frac{3^{k+2}-1}{2}$

$\left(\sum_{i=0}^{k} 3^i\right) + 3^{k+1} = \frac{(3^{k+1})3-1}{2}$

$= \frac{3(3^{k+1})-1-2(3^{k+1})}{2} = \frac{3^{k+1}-1}{2}$

where $n = k+1$

proved

3.) let the distinct power of 2 be denoted of 'n'

let $n = 1$

$\therefore 2^0 + 2^1 = 1 + 2 = 3$ (3 is non -ve integer)

let the statement be true for $n = K$

$\therefore 2^0 + 2^1 + \dots 2^K = N$ (where N is some +ve integer)

let $n = K+1$

$\Rightarrow 2^0 + 2^1 \dots 2^{K+1}$

$\Rightarrow (2^0 + 2^1 \dots 2^K)\, 2^{K+1}$

$\Rightarrow N + 2^{K+1}$ (N is a +ve integer)

adding any +ve no. will result a +ve no

$\therefore$ The statement holds for $n = K+1$

hence proved

4.) let n be the no. of internal node in a full binary tree.

let $n = 1$.

for 1 internal node there can only be 2 leaves.

So, the statements holds good for $n = 1$

let $n = K$

let us assume that the statement holds for $n = K$ internal nodes where there are $R+1$ leaves.

let $n = K+1$

adding 1 internal node to full binary tree will bring 2 leaves.

$\because$ The new internal nodes was a leaf before the no. of lower eaves after adding a new node increase by 1 $(R+2)$

Hence proved

5.) let n be a number

let $n=3$ (3 is divisible by 3)

$n = (10^0 \times 3)$

let $n = abc$ be a no. where all the digits of are divisible by 3.

$n = (10^2 \times a) + (10^1 \times b) + (10^0 * c)$

$= (99+1)a + (l*1)b + c$

$= 99a + 9b + a+b+c$

divisible n by 3.

$\frac{n}{3} = 33a + 3b + \frac{a+b+c}{3}$

$\therefore$ n is divisible by 3 only where sum of digits of n divisible by 3.

7.) $F_{n-1} + F_{n+2}$

let $n=2$

$\therefore F_1 + F_0 \Rightarrow 1+0 = 1$

(given)

let us assume that the f^n that holds for $n=R$

so, $F_{R+1} + F_{K+2}$

$F_{K+1-1} + F_{K+1-2}$

$\Rightarrow F_R + F_{k-1}$

let $R=3$

$F_3 + F_2 \Rightarrow 2+1$

3 is also a fibonaci no. or 4th index

$\therefore$ The statement holds for $n=K+1$ also.

Hence proved.

8.) Let n be the first no of the 3 consecutive natural nos.

let $n=1$

$1^3+2^3+3^3$

$=1+8+27=36$ $\left(\frac{39}{9}=4 \text{ / so divisible by } 9\right)$.

let $n=k$

Let the statement be true for $n=k$

ef $= k^3+(k+1)^3+(k+2)^3$ is divisible by 9 $\langle \lambda \in N)$

let $n=k+1$

$(k+1)^3+(k+2)^3+(k+3)^3$

$\Rightarrow (k+1)^3+(k+2)^3+(k+3)^3$

$\Rightarrow (k+1)^3+(k+2)^3+k^3+27+(9k^2)+(9k)$

$\Rightarrow k^3+(k+1)^3+(k+2)+9(k^2+k+3)$

$\Rightarrow 9\lambda+9(k^2+k+3)$

$\Rightarrow$ The statement hold for $n=k+1$

proved

9.) $\sum_{i=0} (2i+1)^2 = \frac{(n+1)(2n+1)(2n+3)}{3}$.

let $n=0$

$(1)^2 = \frac{(1)(1)(3)}{3} = 1$

Let assume $n=k$ it holds.

So, $\sum_{i=0}^{k} (2i+1)^2 = \frac{(k+1)(2k+1)(2k+3)}{3}$ is true

let $n=k+1$

$\sum_{i=0}^{k} (2i+1)^2 = \frac{(k+2)(2k+2)(2k+5)}{5}$

$1+4+\cdots(2k+1)(2k+2)^2 = \frac{(k+2)(2k+2)(2k+5)}{3}$

$(2k+2)^2 = (2k+2)^2$

$\therefore$ the f^n holds for $n = k+1$ also

Hence proved.

12.) $2^{2x} + 3^{2y+1}$ = prime $(x, y \in N)$

let $x = 1$, $y = 2$

$= 2^{2(1)} + 3^{2(2)+1}$

$= 247$.

but 247 is not prime no $(247 = 13 \times 19)$

$\therefore$ the given statement disproved by counter example.

13.) $\left| \frac{1}{m} - \frac{1}{n} \right| > \frac{1}{2}$ $m \in I$, $n \in N$

let $m = 1$, $n = 1$

$$\left| \frac{1}{1} - \frac{1}{1} \right| = 0 < \frac{1}{2}$$

$\therefore$ The given statement disproved by counter example.

14.) Given n^2 is divisible by 4

let $n^2 = 36$

$\because \frac{36}{4} = 9$ divisible by 4

$n = \sqrt{36} = 6$

$\because \frac{6}{4} = 1.5$ not divisible by 4.

$\therefore$ The given statement disproved by counter example.

15.) $(a+b)^2 = a^2+b^2$

$(2+2)^2 = (2)^2+(2)^3$

$16 \neq 8$.

for $a=3$ and $b=4$

$(3+4)^2 = (3)^2+(4)^2$

$49 \neq 25$.

$\therefore$ The given statement is not algebric identity <u>proved</u>.

16.) $\frac{1}{x+2} = \frac{1}{x} + \frac{1}{2}$

for $x=1$

$\frac{1}{2} = \frac{1}{1} + \frac{1}{2}$

$\frac{1}{2} = \frac{3}{2}$

for $x=2$

$\frac{1}{4} = \frac{1}{2} + \frac{1}{2}$

$\Rightarrow \frac{1}{4} = \frac{1}{4}$

$\therefore$ The statement is not identity <u>proved</u>.

17.) if $pq=x$ $\quad p = \frac{x}{q}$

for $p=3$ $\quad q=2$ $\quad x=6$

$3 \times 2 = 6$

$\therefore 3 = \frac{6}{2} = 3$

for $p=\frac{1}{2}$ $\quad q=2$ $\quad x=1$

$\frac{1}{\cancel{2}} \times \cancel{2} = 1$

$\frac{1}{2} = \frac{1}{2}$

$\therefore$ given statement is true <u>proved</u>

18.) $a+b < \min(a,b)$

for $a=1$ $b=3$

$1+3 < \min(1,3)$

$4 < 1$ which is false

$\therefore$ The given statement is false

proved

19.) x^2 is rational

let $x^2 = 2$

$x = 1.41$ (rational)

let $x^2 = 1.41$

$x = 1.81$ (rational)

$\therefore$ if x^2 is rational x is also rational

proved

20.) $[x+y] = [x] + [y]$

let $x = 0.5$ $y = 0.5$

$[0.5 + 0.5] = [0.5] + [0.5]$

$[1] = 1 + 1$

$1 = 2$ which is false

$\therefore$ by counter example, the following is false.

21.) At the sort of each iteration of the which loop, the sub array $A[n-1 : 1]$ will be conver-ted into file intialization: prior to the first iteration the array A[n-1:1] contains the value 1 so loop works as single time.

maintainance: The while loop works until the given no is greater than 0, converting all no into bits. So until 0, the loop invariant holds for the loop.

Termination: The loop termination when the value of n approacher 0, converting all no. into bits. It given as the binary no. thereby terminating the loop.

22.) The while loop works in the sub array A[0;n-1] thereby comparing all element in an array.

Initialization: The loop invariant holds since there is only one element in an array.

Maintainance: Assume that invariant holds for any Rth iteration ∴ A[1] - A[R-1]. The condn check if the next element is greater or not than A[K-1] if yes it becomes A[1], -, A[K-1], A[R].

Termination:

The loop terminates when all the integer in the array how been checked. Hence we could have our required array after the loop terminates.

24) The while loop works in the subarray A[0; len(A)-1] thereby checking for the required element.

Intialization:- The loop invariant holder, since there is only one element in the array. so it could be our target element.

Maintainance:- There are 3 condition to check wheather the element is in the begining mid or at the end of array. If not found changing mid to some other elements. The loop works until whole array has been reached.

Termination:- The loop terminator when the whole array has been ~~re~~ searched giving us the required element.

25.) ① by mathematical Induction

let $a = 0$

$x^0 = 1$

let assume that x^0 (as if even) if $a \% 2 = 0$

$\therefore$ $a-1$ is odd.

$x(x^{a-1})$

for else condition

if $\frac{a}{2} = 0$ (a is even)

$\therefore$ The algorithm return $x^{2(a/2)}$

$\Rightarrow x^a$

② by loop invariant

```
float power (float x ; int a) {
  while x >= 0
    if (a == 0)
    return 1.0 ;
    else if (a % 2 == 0)
        return x power (x, a-1) ;
    else
      return power (x * x , a/2) ;
    a--;
}
```

Initialization: - The loop invariant holds since a==0 the algorithm returns 1.0

Mantainance :- At the end of every loop, a reduces by 1 computing the no. or per required condn.
∴ The invariant holds until a reduces to zero.

Termination:- The loop terminates when a<0 until a>0 loop works giving us the desired output, with the final output at 1.0.

26.)

a) $f(n) = \frac{n(n-1)}{2}$ $g(n) = 6n$

$\frac{n(n-1)}{2} = O(6n)$

$f(n) = O(g(n))$

b) $f(n) = n + 2\sqrt{n}$ $g(n) = n^2$

$n + 2\sqrt{n} = O\,g(n) = n^2$

$f(n) = O(g(n))$

c) $f(n) = n + \log n$ $g(n) = n\sqrt{n}$

$O(n + \log n) = n\sqrt{n}$

$g(n) = O(f(n))$

d) $f(n) = n \log^n$ $g(n) = \frac{n\sqrt{n}}{2}$

$O(n \log^n) = \frac{n\sqrt{n}}{2}$

$g(n) = O(f(n))$

e) $f(n) = 2(\log n)^2$ $g(n) = \log n + 1$

$2(\log n)^2 = O(\log n + 1)$

$$f(n) = O(g(n))$$

27.) a) $2n^2+1 = O(n^2)$

False.

let $n=2$

$2(2)^2+1=9$

$(2)^2=4$

b) $n^2(1+\sqrt{n}) = O(n^2)$

False.

let $n=4$ $\quad (4)^2(1+2)=48$

$(4)^2=16$.

c) $n^2(1+\sqrt{n}) = O(n^2 \log n)$

False.

let $n=4$

$(4)^2(1+2)=48$

$(4^2\log 4) = 2218$

d) $3n^2+\sqrt{n} = O(n+n\sqrt{n}+\sqrt{n})$

let $n=4$.

$3(4)^2+2=50$

$4+4(2)+2=14$

e) $\sqrt{n}\log n = O(n)$

let $n=4$

$2\log(4) = 2.77 \quad n=4$

False

f) $\log(n) \in O(n)$

True

g) $n \in O(n\log n)$

True

h) $n \log n \in O(n^2)$

True

i) $2^n \in O(6^{n \log n})$

True

j) $\log^3 n \in O(n^{0.5})$

False

28.) a) $f(n) = \sqrt{n}$ $\quad g(n) = \log(n+3)$

let $n = 4$

$\sqrt{4} = 2$

$\log(7) = 0.84$

$\therefore f(n) = \Omega(g(n))$

b) $f(n) = n\sqrt{n}$ $\quad g(n) = n^2 - n$

let $n = 4$

$(4)(2) = 8$

$16 - 4 = 12$ $\qquad \therefore f(n) = O(g(n))$

c) $f(n) = 2^n - n^2$ $\qquad g(n) = n^4 + n^2$

let $n = 5$

$2^5 - 5^2 = 7$

$5^4 - 5^2 = 650$

$\therefore f(n) = O(g(n))$

d) $f(n) = n^2 + 3n + 4$ $\qquad g(n) = 6n^2$

let $n = 2$

$(2)^2 + (3(2) + 4 = 14$

$6(2)^2 = 24$

$f(n) = \Omega(g(n))$

e) $F(n) = n + n\sqrt{n}$ $\qquad g(n) = 4^n(\log(4^2+1)$

let $n = 4$

$(4 + 4\sqrt{4}) = 12$ $\qquad 4(4)\log(10+1) = 19.68$

$\therefore f(n) = O(g(n))$

29.) a) $f_1(n) = \Omega(g_1(n))$

$f_2(n) = \Omega(g_2(n))$

$0 \leq c_1 g_1(n) \leq f(n)$ —— (1)

$0 \leq c_2 g_2(n) \leq f_2(n)$ —— (2)

adding (1) & (2)

$0 \leq c_3 (g_1(n) + g_2(n)) \leq f_1(n) + f_2(n)$

$\Rightarrow \quad f_1(n) + f_2(n) = \Omega(g_1(n) + g_2(n))$

proved

c) $f_1(n) = O(g_1(n))$

$f_2(n) = O(g_2(n))$

$0 \leq f_1(n) \leq c_1 (g_1(n))$ —— (1)

$0 \leq f_2(n) \leq c_2 (g_2(n))$ —— (2)

$\therefore$ (1) & (2)

$0 \leq f_1(n) \times f_2(n) \leq c_3 [g_1(n) \times g_2(n)]$

d) $f_n = O[g(n)]$

$0 \leq f(n) \leq c(g(n))$

Let R be some constant.

$0 \leq f(n)^R \leq c^R \cdot g(n)^R$

$\therefore f(n)^R = O[g(n)^R]$ proved

31.) Given $b > a > 0$

$\therefore b^n > a^n$

So, we can say b^n takes more time to compute as a^n.

$\therefore a^n \in O(b^n)$ proved

32.) To prove: $\log(n) = O(\sqrt{n})$

we prove:

$0 \leq \log n \leq c \cdot \sqrt{n}$

let $n = 4$

$0 \leq 0.602 \leq c \cdot 2$

which is true

$\therefore \log n = O(\sqrt{n})$ proved

35.)
```
function (int n) {
    if (n==1)
        return 1
    else
        function(n/3) ; function(n/3);
                        function(n/3);
        for (i=1; i<=n; i++)
            x = x+1
}
```

Time	space
1.	4 bytes
1.	
1.	
$3(2^n)$	
n	n bytes.
$n-1$	

$$T(n) = 3 + 3(2^n) + n + (n-1)$$
$$= 3 + 3(2^n) + 2n - 1$$
$$= 3(2^n) + 2^n + 2 \quad = O(\underline{3(2^n)} + 1(n+1))$$

$$T(n) = O(5(2^n))$$

space complex $= O(n^2)$.

36.) void function (int n) {

Code	Time C	space
temp = 1;	1	4 bytes
repeat	n+1	n bytes
for i = 1 to n	n	n
temp = temp + 1;	n-1	4 bytes
n = n/2;		
until n <= 1	n-1	4 bytes

}

$$T(n) = C_1(1) + C_2(n+1) + C_3(n) + C_4(n-1) + C_5(n-1)$$

$T(n) = \underline{O(4n)}$ Time complexity.

Space Complex. $= 12 + n$

$$= O(n)$$

37.) int function (int n) {

```
    if (n <= 2)
        return 1;
    else
        return (function (floor((sqrt(n))+1));
}
```

38.)

```
void function(int n) {
  if (n==1)
     return 1;
  else
    for (i=1; i<=8 ; i++)
          function (n/2);
    for (i=1; i<=n^3 ; i++ )
        Count = count+1;
  }
```

Time (C)	Space.
1	4 bytes
9 / 2^n	4 bytes / $4n+2$
n^3	4 bytes
n^3-1	4 bytes

$$T(n) = 1+9+2^n+n^3+(n^3-1)$$

$$= 9+2^n+2n^3$$

$$T(n) = T(2^n)+O(n^3)$$

Space C : $= 16+4n+2$

$= 18+4n$

$= O(5n)$

39.) Recursion version of L.Search

```
int linear_search (int arr[], int r, int x, int l)
 {
   if (r<1)
       return -1
   else if ( arr[i] == x)
          return i ;
   else if (arr[r] == x)
          return r;
   return linear-search (arr, l+1, r-1, x);
 }
```

$T(n) = T(2^n) + O(n)$

Space comp = $O(1)$

33)

10	11						70
0	1						60

<u>In binary search Algorithm</u> :—

(1) If the element is in middle no comparision are needed and element found at once.

(2) If element is at begining or at end array. The would be total <u>7 comparision</u> for each of the cases, since the algorithm divide the array in 2 parts; minimum <u>7 comparision</u> would be needed.