DESIGN AND ANALYSIS OF ALGORITHMS

EXPERIMENT 1A

ANUSHKA ACHARYA
CSE DS D1 BATCH
UID: 2021700001

AIM:
TO IMPLEMENT VARIOUS FUNCTIONS E.G. LINEAR, NON-LINEAR, QUADRATIC, EXPONENTIAL, ETC.

THEORY:
A function is a process or a relation that associates each element 'a' of a non-empty set A , at least to a single element 'b' of another non-empty set B.
A relation f from a set A (the domain of the function) to another set B (the co-domain of the function) is called a function in math. f = {(a,b)| for all a ∈ A, b ∈ B}

- A relation is said to be a function if every element of set A has one and only one image in set B.
- A function is a relation from a non-empty set B such that the domain of a function is A and no two distinct ordered pairs in f have the same first element.
- A function from A → B and (a,b) ∈ f, then f(a) = b, where 'b' is the image of 'a' under 'f' and 'a' is the preimage of 'b' under 'f'.
- If there exists a function f: A → B, the set A is called the domain of the function f, and the set B is called its co-domain.

ALGORITHM:

STEP_1: Start.
STEP_2: Print the header row to make one column for input, ten columns for functions and last column for factorial of a number.
STEP_3: Start a for loop which will run from i=0 to 100 to find values of functions for all numbers from 0 to 100. In this loop:
(1) First print the input.
(2) Define function f1 as (3/2)^n. Print its value.
(3) Define function f2 as n^3. Print its value.
(4) Define function f3 as 2^(2^n). Print its value.
(5) Define function f4 as ln(ln(n)). Print its value.
(6) Define function f5 as n*(2^n). Print its value.
(7) Define function f6 as ln(n). Print its value.
(8) Define function f7 as 2^n. Print its value.
(9) Define function f8 as 2^(2^(n+1)). Print its value.
(10) Define function f9 as e^n. Print its value.
(11) Define function f10 as lg(n). Print its value.
(12) To print the factorial of first 20 numbers, put an if statement. Initialise fact=1, put fact=fact*i to calculate factorial and print its value.
STEP_4: Stop.

CODE:

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

int
main ()
{
  printf
    ("Input (3/2)^n       n^3      2^(2^n)          ln(ln(n))          n*(2^n)          ln(n)    2^n
          2^(2^(n+1))    e^n      lg(n)    Factorial\n");

  for (int i = 0; i <= 100; i++)
    {
      printf ("%d ", i);
      float f1 = pow ((float) 3 / 2, i);
      printf ("%f ", f1);
      float f2 = pow (i, 3);
      printf ("%f ", f2);
      float f3 = pow (2, pow (2, i));
      printf ("%f ", f3);
      float f4 = log (log (i));
      printf ("%f ", f4);
      float f5 = i * pow (2, i);
      printf ("%f ", f5);
      float f6 = log (i);
      printf ("%f ", f6);
      float f7 = pow (2, i);
      printf ("%f ", f7);
      float f8 = pow (2, pow (2, i + 1));
      printf ("%f ", f8);
      float f9 = pow (2.71, i);
      printf ("%f ", f9);
      float f10 = log (i) / log (2);
      printf ("%f ", f10);
      int fact = 1;
      if (i == 0)
           printf ("0 ");
      else if (i <= 20)
           {
             fact *= i;
             printf ("%d ", fact);
           }
      printf ("\n");
    }

  return 0;
}
```

OUTPUT:

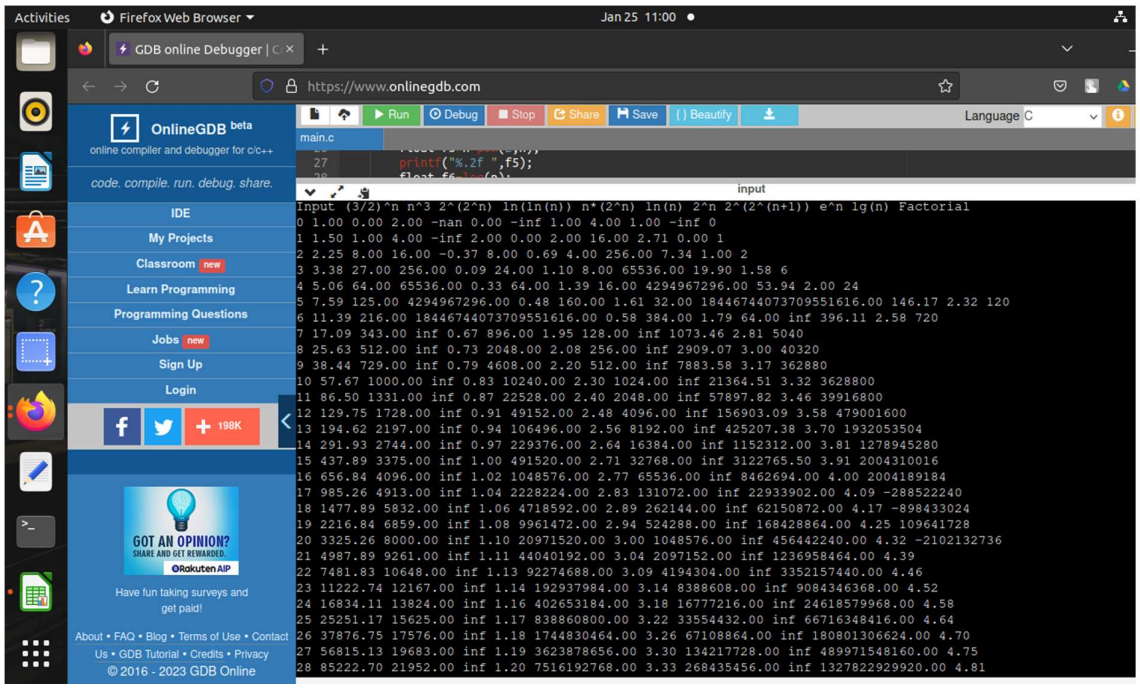


TABLE:

Input	(3/2)^n	n^3	2^(2^n)	ln(ln(n))	n*(2^n)	ln(n)	2^n	2^(2^(n+1))	e^n	lg(n)	Factorial
0	1	0	2	-nan	0	-inf	1	4	1	-inf	0
1	1.5	1	4	-inf	2	0	2	16	2.71	0	1
2	2.25	8	16	-0.37	8	0.69	4	256	7.34	1	2
3	3.38	27	256	0.09	24	1.1	8	65536	19.9	1.58	6
4	5.06	64	65536	0.33	64	1.39	16	4294967296	53.94	2	24
5	7.59	125	4294967296	0.48	160	1.61	32	1.8447E+19	146.17	2.32	120
6	11.39	216	1.8447E+19	0.58	384	1.79	64	inf	396.11	2.58	720
7	17.09	343	inf	0.67	896	1.95	128	inf	1073.46	2.81	5040
8	25.63	512	inf	0.73	2048	2.08	256	inf	2909.07	3	40320
9	38.44	729	inf	0.79	4608	2.2	512	inf	7883.58	3.17	362880
10	57.67	1000	inf	0.83	10240	2.3	1024	inf	21364.51	3.32	3628800
11	86.5	1331	inf	0.87	22528	2.4	2048	inf	57897.82	3.46	39916800
12	129.75	1728	inf	0.91	49152	2.48	4096	inf	156903.09	3.58	479001600
13	194.62	2197	inf	0.94	106496	2.56	8192	inf	425207.38	3.7	1932053504
14	291.93	2744	inf	0.97	229376	2.64	16384	inf	1152312	3.81	1278945280
15	437.89	3375	inf	1	491520	2.71	32768	inf	3122765.5	3.91	2004310016

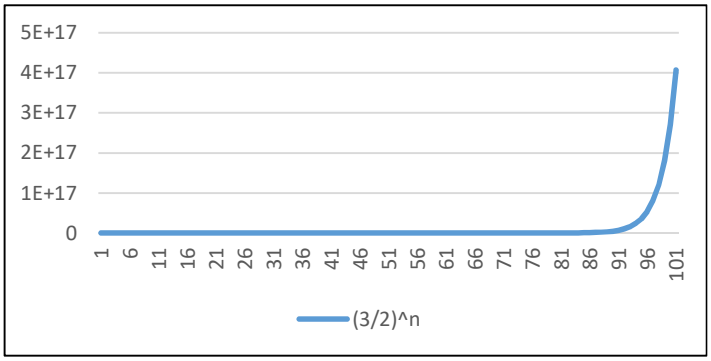
16	656.84	4096	inf	1.02	1048576	2.77	65536	inf	8462694	4	2004189184
17	985.26	4913	inf	1.04	2228224	2.83	131072	inf	2293902	4.09	-2885222240
18	1477.89	5832	inf	1.06	4718592	2.89	262144	inf	6215072	4.17	-8984330240
19	2216.84	6859	inf	1.08	9961472	2.94	524288	inf	1684288 64	4.25	1096417280
20	3325.26	8000	inf	1.1	20971520	3	1048576	inf	4564424 0	4.32	-2.102E+09
21	4987.89	9261	inf	1.11	44040192	3.04	2097152	inf	1236958464	4.39	
22	7481.83	10648	inf	1.13	92274688	3.09	4194304	inf	3352157440	4.46	
23	11222.74	12167	inf	1.14	192937984	3.14	8388608	inf	9084346368	4.52	
24	16834.11	13824	inf	1.16	402653184	3.18	16777216	inf	2.4619E+10	4.58	
25	25251.17	15625	inf	1.17	838860800	3.22	33554432	inf	6.6716E+10	4.64	
26	37876.75	17576	inf	1.18	1744830464	3.26	67108864	inf	1.808E+11	4.7	
27	56815.13	19683	inf	1.19	3623878656	3.3	134217728	inf	4.8997E+11	4.75	
28	85222.7	21952	inf	1.2	7516192768	3.33	268435456	inf	1.3278E+12	4.81	
29	127834.04	24389	inf	1.21	1.5569E+10	3.37	536870912	inf	3.5984E+12	4.86	
30	191751.06	27000	inf	1.22	3.2212E+10	3.4	1073741824	inf	9.7517E+12	4.91	
31	287626.59	29791	inf	1.23	6.6572E+10	3.43	2147483648	inf	2.6427E+13	4.95	
32	431439.88	32768	inf	1.24	1.3744E+11	3.47	4294967296	inf	7.1617E+13	5	
33	647159.81	35937	inf	1.25	2.8347E+11	3.5	8589934592	inf	1.9408E+14	5.04	
34	970739.75	39304	inf	1.26	5.8412E+11	3.53	1.718E+10	inf	5.2596E+14	5.09	
35	1456109.62	42875	inf	1.27	1.2026E+12	3.56	3.436E+10	inf	1.4254E+15	5.13	
36	2184164.5	46656	inf	1.28	2.4739E+12	3.58	6.8719E+10	inf	3.8627E+15	5.17	
37	3276246.5	50653	inf	1.28	5.0852E+12	3.61	1.3744E+11	inf	1.0468E+16	5.21	
38	4914370	54872	inf	1.29	1.0445E+13	3.64	2.7488E+11	inf	2.8368E+16	5.25	
39	7371555	59319	inf	1.3	2.144E+13	3.66	5.4976E+11	inf	7.6878E+16	5.29	
40	11057332	64000	inf	1.31	4.398E+13	3.69	1.0995E+12	inf	2.0834E+17	5.32	
41	16585998	68921	inf	1.31	9.016E+13	3.71	2.199E+12	inf	5.646E+17	5.36	
42	24878998	74088	inf	1.32	1.8472E+14	3.74	4.398E+12	inf	1.5301E+18	5.39	
43	37318496	79507	inf	1.32	3.7823E+14	3.76	8.7961E+12	inf	4.1465E+18	5.43	
44	55977744	85184	inf	1.33	7.7406E+14	3.78	1.7592E+13	inf	1.1237E+19	5.46	
45	83966616	91125	inf	1.34	1.5833E+15	3.81	3.5184E+13	inf	3.0452E+19	5.49	
46	125949928	97336	inf	1.34	3.237E+15	3.83	7.0369E+13	inf	8.2525E+19	5.52	
47	188924896	103823	inf	1.35	6.6147E+15	3.85	1.4074E+14	inf	2.2364E+20	5.55	
48	283387328	110592	inf	1.35	1.3511E+16	3.87	2.8147E+14	inf	6.0607E+20	5.58	
49	425080992	117649	inf	1.36	2.7585E+16	3.89	5.6295E+14	inf	1.6425E+21	5.61	
50	637621504	125000	inf	1.36	5.6295E+16	3.91	1.1259E+15	inf	4.4511E+21	5.64	

51	956432256	132651	inf	1.37	1.1484E+17	3.93	2.2518E+15	inf	1.2062E+22	5.67	
52	1434648320	140608	inf	1.37	2.3419E+17	3.95	4.5036E+15	inf	3.2689E+22	5.7	
53	2151972608	148877	inf	1.38	4.7738E+17	3.97	9.0072E+15	inf	8.8587E+22	5.73	
54	3227958784	157464	inf	1.38	9.7278E+17	3.99	1.8014E+16	inf	2.4007E+23	5.75	
55	4841938432	166375	inf	1.39	1.9816E+18	4.01	3.6029E+16	inf	6.506E+23	5.78	
56	7262907392	175616	inf	1.39	4.0352E+18	4.03	7.2058E+16	inf	1.7631E+24	5.81	
57	1.0894E+10	185193	inf	1.4	8.2146E+18	4.04	1.4412E+17	inf	4.778E+24	5.83	
58	1.6342E+10	195112	inf	1.4	1.6717E+19	4.06	2.8823E+17	inf	1.2948E+25	5.86	
59	2.4512E+10	205379	inf	1.41	3.4011E+19	4.08	5.7646E+17	inf	3.509E+25	5.88	
60	3.6768E+10	216000	inf	1.41	6.9175E+19	4.09	1.1529E+18	inf	9.5095E+25	5.91	
61	5.5153E+10	226981	inf	1.41	1.4066E+20	4.11	2.3058E+18	inf	2.5771E+26	5.93	
62	8.2729E+10	238328	inf	1.42	2.8592E+20	4.13	4.6117E+18	inf	6.9839E+26	5.95	
63	1.2409E+11	250047	inf	1.42	5.8107E+20	4.14	9.2234E+18	inf	1.8926E+27	5.98	
64	1.8614E+11	262144	inf	1.43	1.1806E+21	4.16	1.8447E+19	inf	5.129E+27	6	
65	2.7921E+11	274625	inf	1.43	2.3981E+21	4.17	3.6893E+19	inf	1.39E+28	6.02	
66	4.1882E+11	287496	inf	1.43	4.8699E+21	4.19	7.3787E+19	inf	3.7668E+28	6.04	
67	6.2822E+11	300763	inf	1.44	9.8875E+21	4.2	1.4757E+20	inf	1.0208E+29	6.07	
68	9.4234E+11	314432	inf	1.44	2.007E+22	4.22	2.9515E+20	inf	2.7664E+29	6.09	
69	1.4135E+12	328509	inf	1.44	4.073E+22	4.23	5.903E+20	inf	7.4969E+29	6.11	
70	2.1203E+12	343000	inf	1.45	8.2641E+22	4.25	1.1806E+21	inf	2.0317E+30	6.13	
71	3.1804E+12	357911	inf	1.45	1.6764E+23	4.26	2.3612E+21	inf	5.5058E+30	6.15	
72	4.7706E+12	373248	inf	1.45	3.4001E+23	4.28	4.7224E+21	inf	1.4921E+31	6.17	
73	7.1559E+12	389017	inf	1.46	6.8947E+23	4.29	9.4447E+21	inf	4.0435E+31	6.19	
74	1.0734E+13	405224	inf	1.46	1.3978E+24	4.3	1.8889E+22	inf	1.0958E+32	6.21	
75	1.6101E+13	421875	inf	1.46	2.8334E+24	4.32	3.7779E+22	inf	2.9696E+32	6.23	
76	2.4151E+13	438976	inf	1.47	5.7424E+24	4.33	7.5558E+22	inf	8.0476E+32	6.25	
77	3.6227E+13	456533	inf	1.47	1.1636E+25	4.34	1.5112E+23	inf	2.1809E+33	6.27	
78	5.434E+13	474552	inf	1.47	2.3574E+25	4.36	3.0223E+23	inf	5.9102E+33	6.29	
79	8.151E+13	493039	inf	1.47	4.7753E+25	4.37	6.0446E+23	inf	1.6017E+34	6.3	
80	1.2226E+14	512000	inf	1.48	9.6714E+25	4.38	1.2089E+24	inf	4.3405E+34	6.32	
81	1.834E+14	531441	inf	1.48	1.9585E+26	4.39	2.4179E+24	inf	1.1763E+35	6.34	
82	2.751E+14	551368	inf	1.48	3.9653E+26	4.41	4.8357E+24	inf	3.1877E+35	6.36	
83	4.1264E+14	571787	inf	1.49	8.0273E+26	4.42	9.6714E+24	inf	8.6388E+35	6.38	
84	6.1896E+14	592704	inf	1.49	1.6248E+27	4.43	1.9343E+25	inf	2.3411E+36	6.39	
85	9.2845E+14	614125	inf	1.49	3.2883E+27	4.44	3.8686E+25	inf	6.3444E+36	6.41	
86	1.3927E+15	636056	inf	1.49	6.6539E+27	4.45	7.7371E+25	inf	1.7193E+37	6.43	

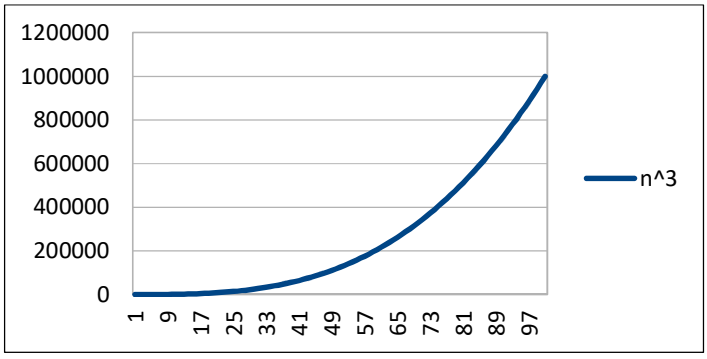
87	2.089E+15	658503	inf	1.5	1.3463E+28	4.47	1.5474E+26	inf	4.6594E+37	6.44	
88	3.1335E+15	681472	inf	1.5	2.7235E+28	4.48	3.0949E+26	inf	1.2627E+38	6.46	
89	4.7003E+15	704969	inf	1.5	5.5088E+28	4.49	6.1897E+26	inf	inf	6.48	
90	7.0504E+15	729000	inf	1.5	1.1141E+29	4.5	1.2379E+27	inf	inf	6.49	
91	1.0576E+16	753571	inf	1.51	2.2531E+29	4.51	2.4759E+27	inf	inf	6.51	
92	1.5863E+16	778688	inf	1.51	4.5556E+29	4.52	4.9518E+27	inf	inf	6.52	
93	2.3795E+16	804357	inf	1.51	9.2103E+29	4.53	9.9035E+27	inf	inf	6.54	
94	3.5693E+16	830584	inf	1.51	1.8619E+30	4.54	1.9807E+28	inf	inf	6.55	
95	5.3539E+16	857375	inf	1.52	3.7633E+30	4.55	3.9614E+28	inf	inf	6.57	
96	8.0308E+16	884736	inf	1.52	7.6059E+30	4.56	7.9228E+28	inf	inf	6.58	
97	1.2046E+17	912673	inf	1.52	1.537E+31	4.57	1.5846E+29	inf	inf	6.6	
98	1.8069E+17	941192	inf	1.52	3.1057E+31	4.58	3.1691E+29	inf	inf	6.61	
99	2.7104E+17	970299	inf	1.52	6.2749E+31	4.6	6.3383E+29	inf	inf	6.63	
100	4.0656E+17	1000000	inf	1.53	1.2677E+32	4.61	1.2677E+30	inf	inf	6.64	

GRAPHS:

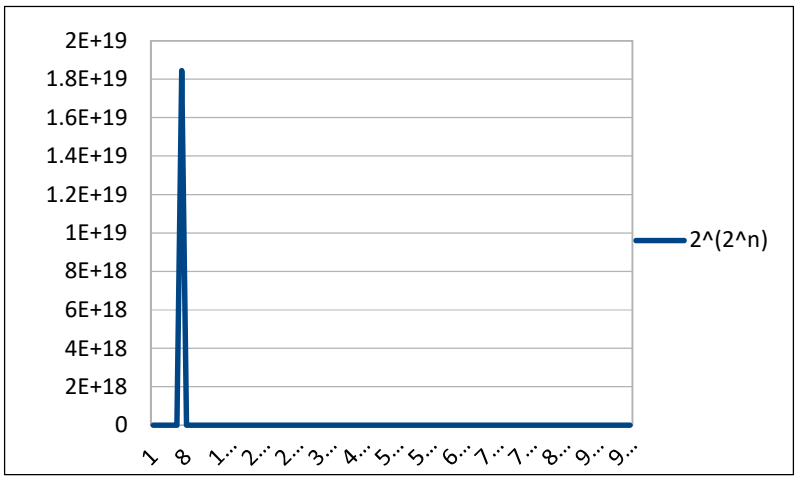
OBSERVATION_(1) (3/2)^n: The values of the function are gradually increasing. At n=83 there is a sudden rise in value after which the result tends to infinity.



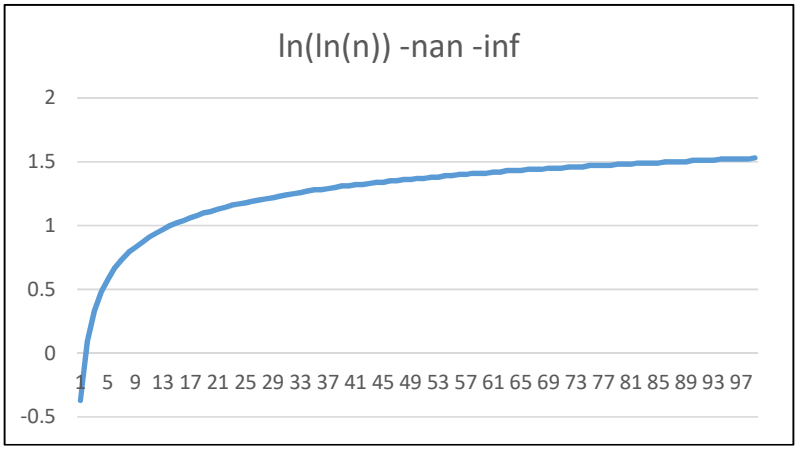
OBSERVATION_(2) n^3: This function has a U shaped graph which starts from 0 and gradually increases to infinity.



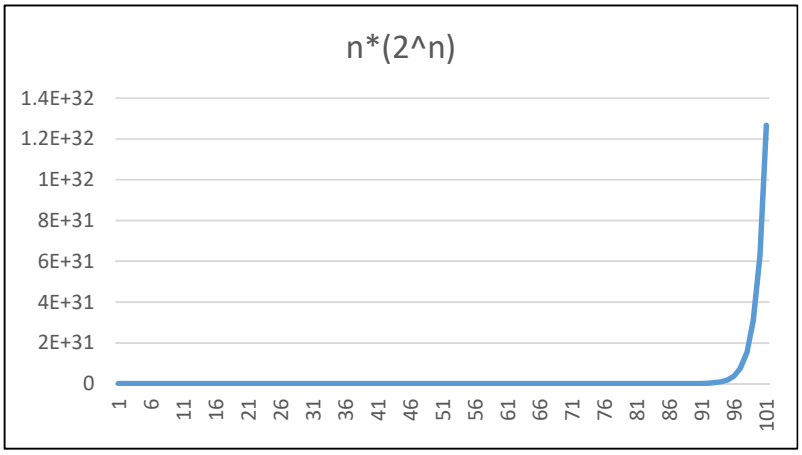
OBSERVATION_(3) 2^(2^n): After n=8, the graph of this functions tends to infinity.



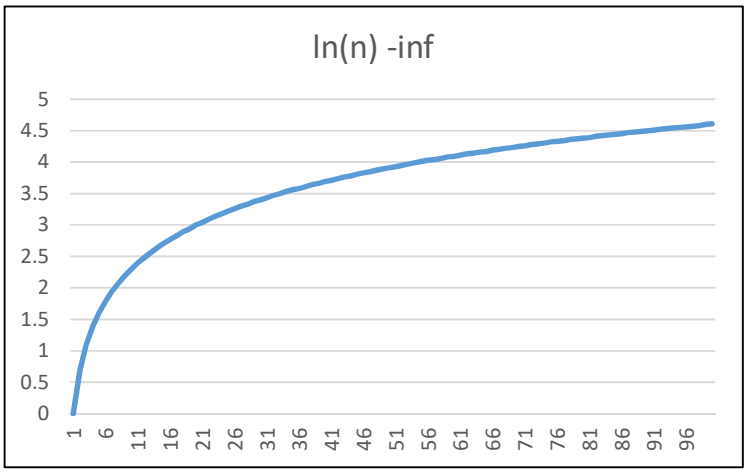
OBSERVATION_(4) ln(ln(n)): This function has a negative value at n=2. The graph has sudden increase at first but then gradually acquires a lesser slope.



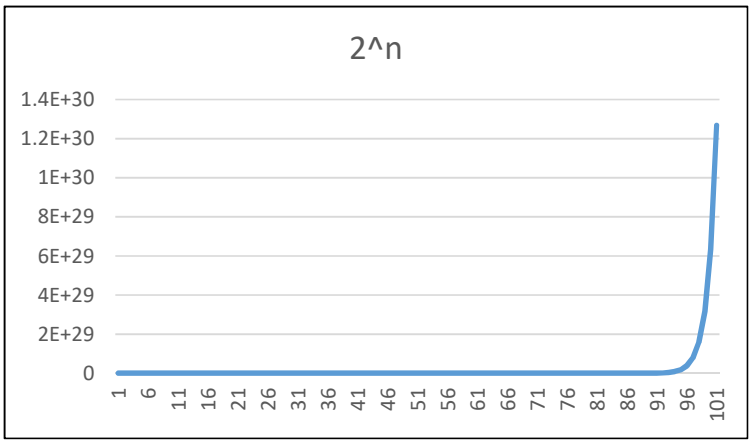
OBSERVATION_(5) n*(2^n): This function has a sudden rise in value at n=92 after which it tends to infinity.



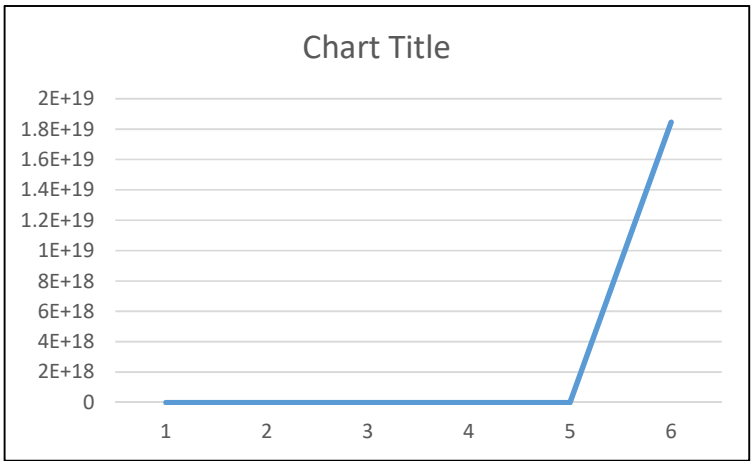
OBSERVATION_(6) ln(n): The graph starts from ln(1)=0. It first increases steeply and then becomes gradual for higher values of n.



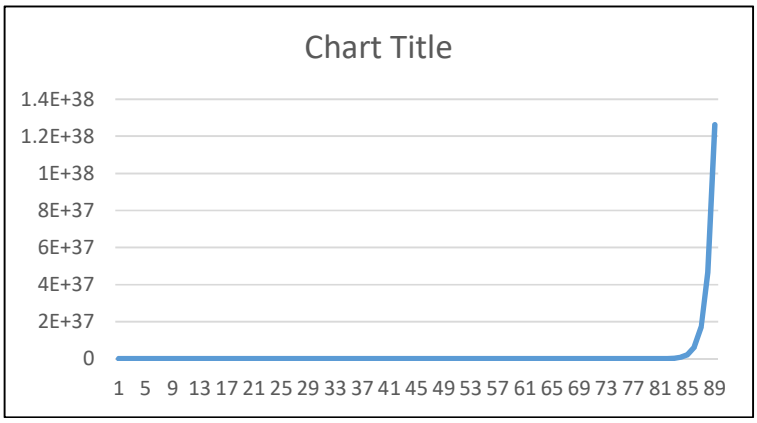
OBSERVATION_(7) 2^n: This function has a sudden rise in value at n=92 after which it tends to infinity.



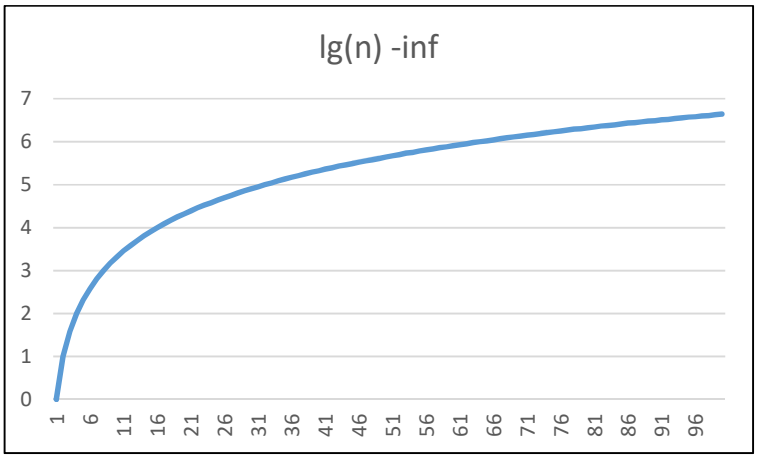
OBSERVATION_(8) 2^(2^(n+1)): The graph has gradual increase in its values from n=0 to 5, at n=5 there is a sudden increase and the function gradually tends to infinity.



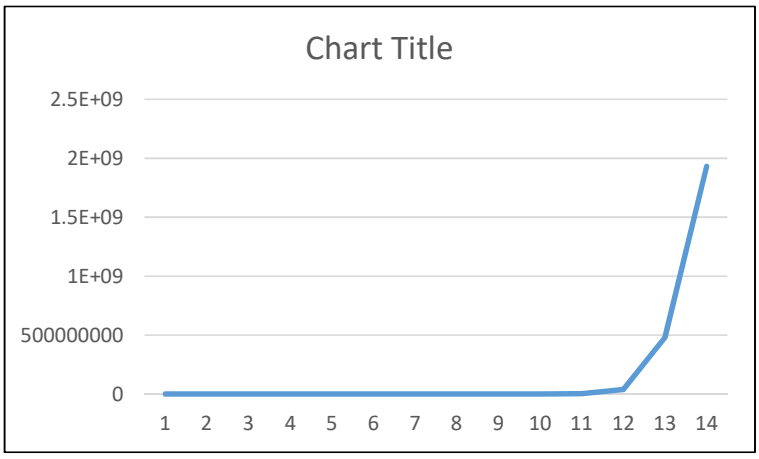
OBSERVATION_(9) e^n: The values of the function are gradually increasing. At n=83 there is a sudden rise in value after which the result tends to infinity.



OBSERVATION_(10) lg(n): The graph starts from lg(1)=0. It first increases steeply and then becomes gradual for higher values of n.



OBSERVATION_(11) Factorial: The value of factorial keeps on increasing as n increases. There is a sudden rise in value at n=13.



CONCLUSION:
From this experiment I learnt how to implement various functions in C Programming language for values of n varying from 0 to 100, and also understood how the graph of each function is affected as value of n changes.