Name:	Prithvi Singh
UID:	2022301014
Subject:	Design and Analysis of Algorithm
Experiment No. :	1A
Aim:	To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.
Algorithm:	Function 1: i. Initialize a variable n. ii. Take the value of n from 0-100 and print all of them. Function 2: i. Initialize variables n and result. ii. result = n*n*n iii. Apply a for loop for values of n from 0-100 and print all the values for result. Function 3: i. Initialize variables n and result. ii. result = (3/2)^n iii. Apply a for loop for values of n from 0-100 and print all the values for result. Function 4: i. Initialize variables n and result. ii. result = log10(n) iii. Apply a for loop for values of n from 0-100 and print all the values for result. Function 5: i. Initialize variables n and result. ii. result = n*(2)^n iii. Apply a for loop for values of n from 0-100 and print all the values for result. Function 6: i. Initialize variables n and result. ii. result = 2^logn

	iii. Apply a for loop for values of n from 0-100 and print all the values for result.
	
	Function 7:
	i. Initialize variables n and result.
	ii. result = 2^n
	iii. Apply a for loop for values of n from 0-100 and print all the values for result.
	
	Function 8:
	i. Initialize variables n and result.
	ii. result = e^n
	iii. Apply a for loop for values of n from 0-100 and print all the values for result.
	
	Function 9:
	iv. Initialize variables n and result.
	v. result = log2n
	vi. Apply a for loop for values of n from 0-100 and print all the values for result.
	
	Function 10:
	i. Initialize variables n and result.
	ii. result = ln(n)
	iii. Apply a for loop for values of n from 0-100 and print all the values for result.
	
	Function 12:
	i. Initialize a variable n
	ii. Create a function to find the factorial.
	iii. factorial(n)
	if(n==1 || n==0)returni
	else
	return n*factorial(n-1)
	Apply a for loop for values of n from 0-19 and print al thevalues for result in the main function.

Code:

```
#include <iostream>
#include <math.h>
#include<fstream>
using namespace std;


double ncube(int n){
  return pow(n,3);
}
double threebytwon(int n){
  return pow((3.0/2.0),n);
}

double logn(int n){
  return log(n)/log(2);
}

double ntwon(int n){
  return n*pow(2,n);
}

double twopowerlogn(int n){
  return pow(2,logn(n));
}

double tworaiseton(int n){
  return pow(2,n);
}

double eRaisedToN(int n){
    return exp(n);
}

double lognsquare(int n){
  return pow(logn(n),2);
}

double inN(int n){
  return log(n);
}

double nfunction(int n){
  return n;
}

float factorial(int a){
  if(a<=1){
    return 1;
  }
```

```
 return a * factorial(a-1);
}

int main() {
 int n,a;
 ofstream myFile;
 myFile.open("data.txt");
 myFile<<"Hello"<<endl;
 myFile.close();


  ofstream myfile("file.csv");

 for(int i=1;i<=100;i++){
  myfile<<i<<","<<
  ncube(i)<<","<<
  threebytwon(i)<<","<<
  logn(i)<<","<<
  ntwon(i)<<","<<
  twopowerlogn(i)<<","<<
  tworaiseton(i)<<","<<
  eRaisedToN(i)<<","<<
  lognsquare(i)<<","<<
  inN(i)<<","<<
  nfunction(i)<<endl;

}
 myfile.close();

 int i=1;

 while(i<=20){
  float a = factorial(i);
  cout<<i<<"-"<<a<<endl;
  i++;
 }
 return 0;
 }
```

Observation:

n	n^3	$(3/2)^n$	log n	$n*2^n$	$2^{\log n}$	2^n	e^n	$\log^2 n$	In n	n
1	1	1.5	0	2	1	2	2.71828	0	0	1
2	8	2.25	1	8	2	4	7.38906	1	0.693147	2
3	27	3.375	1.58496	24	3	8	20.0855	2.51211	1.09861	3
4	64	5.0625	2	64	4	16	54.5982	4	1.38629	4
5	125	7.59375	2.32193	160	5	32	148.413	5.39135	1.60944	5
6	216	11.3906	2.58496	384	6	64	403.429	6.68203	1.79176	6
7	343	17.0859	2.80735	896	7	128	1096.63	7.88124	1.94591	7
8	512	25.6289	3	2048	8	256	2980.96	9	2.07944	8
9	729	38.4434	3.16993	4608	9	512	8103.08	10.0484	2.19722	9
10	1000	57.665	3.32193	10240	10	1024	22026.5	11.0352	2.30259	10
11	1331	86.4976	3.45943	22528	11	2048	59874.1	11.9677	2.3979	11
12	1728	129.746	3.58496	49152	12	4096	162755	12.852	2.48491	12
13	2197	194.62	3.70044	106496	13	8192	442413	13.6933	2.56495	13
14	2744	291.929	3.80735	229376	14	16384	1.20E+06	14.496	2.63906	14
15	3375	437.894	3.90689	491520	15	32768	3.27E+06	15.2638	2.70805	15
16	4096	656.841	4	1.05E+06	16	65536	8.89E+06	16	2.77259	16
17	4913	985.261	4.08746	2.23E+06	17	131072	2.42E+07	16.7074	2.83321	17
18	5832	1477.89	4.16993	4.72E+06	18	262144	6.57E+07	17.3883	2.89037	18
19	6859	2216.84	4.24793	9.96E+06	19	524288	1.78E+08	18.0449	2.94444	19
20	8000	3325.26	4.32193	2.10E+07	20	1.05E+06	4.85E+08	18.6791	2.99573	20
21	9261	4987.89	4.39232	4.40E+07	21	2.10E+06	1.32E+09	19.2925	3.04452	21
22	10648	7481.83	4.45943	9.23E+07	22	4.19E+06	3.58E+09	19.8865	3.09104	22
23	12167	11222.7	4.52356	1.93E+08	23	8.39E+06	9.74E+09	20.4626	3.13549	23
24	13824	16834.1	4.58496	4.03E+08	24	1.68E+07	2.65E+10	21.0219	3.17805	24
25	15625	25251.2	4.64386	8.39E+08	25	3.36E+07	7.20E+10	21.5654	3.21888	25
26	17576	37876.8	4.70044	1.74E+09	26	6.71E+07	1.96E+11	22.0941	3.2581	26
27	19683	56815.1	4.75489	3.62E+09	27	1.34E+08	5.32E+11	22.609	3.29584	27
28	21952	85222.7	4.80735	7.52E+09	28	2.68E+08	1.45E+12	23.1107	3.3322	28
29	24389	127834	4.85798	1.56E+10	29	5.37E+08	3.93E+12	23.6	3.3673	29
30	27000	191751	4.90689	3.22E+10	30	1.07E+09	1.07E+13	24.0776	3.4012	30
31	29791	287627	4.9542	6.66E+10	31	2.15E+09	2.90E+13	24.5441	3.43399	31
32	32768	431440	5	1.37E+11	32	4.29E+09	7.90E+13	25	3.46574	32
33	35937	647160	5.04439	2.83E+11	33	8.59E+09	2.15E+14	25.4459	3.49651	33
34	39304	970740	5.08746	5.84E+11	34	1.72E+10	5.83E+14	25.8823	3.52636	34
35	42875	1.46E+06	5.12928	1.20E+12	35	3.44E+10	1.59E+15	26.3095	3.55535	35
36	46656	2.18E+06	5.16993	2.47E+12	36	6.87E+10	4.31E+15	26.7281	3.58352	36
37	50653	3.28E+06	5.20945	5.09E+12	37	1.37E+11	1.17E+16	27.1384	3.61092	37
38	54872	4.91E+06	5.24793	1.04E+13	38	2.75E+11	3.19E+16	27.5407	3.63759	38
39	59319	7.37E+06	5.2854	2.14E+13	39	5.50E+11	8.66E+16	27.9355	3.66356	39
40	64000	1.11E+07	5.32193	4.40E+13	40	1.10E+12	2.35E+17	28.3229	3.68888	40
41	68921	1.66E+07	5.35755	9.02E+13	41	2.20E+12	6.40E+17	28.7034	3.71357	41
42	74088	2.49E+07	5.39232	1.85E+14	42	4.40E+12	1.74E+18	29.0771	3.73767	42

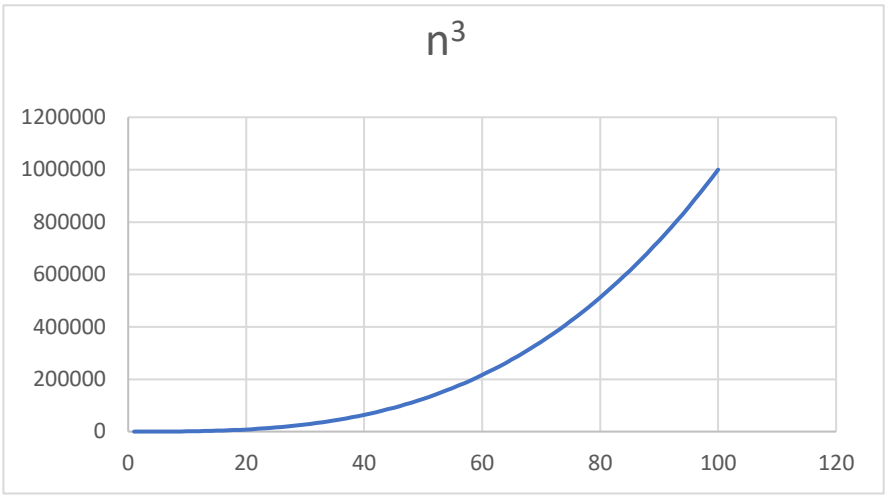
43	79507	3.73E+07	5.42626	3.78E+14	43	8.80E+12	4.73E+18	29.4443	3.7612	43
44	85184	5.60E+07	5.45943	7.74E+14	44	1.76E+13	1.29E+19	29.8054	3.78419	44
45	91125	8.40E+07	5.49185	1.58E+15	45	3.52E+13	3.49E+19	30.1605	3.80666	45
46	97336	1.26E+08	5.52356	3.24E+15	46	7.04E+13	9.50E+19	30.5097	3.82864	46
47	103823	1.89E+08	5.55459	6.61E+15	47	1.41E+14	2.58E+20	30.8535	3.85015	47
48	110592	2.83E+08	5.58496	1.35E+16	48	2.81E+14	7.02E+20	31.1918	3.8712	48
49	117649	4.25E+08	5.61471	2.76E+16	49	5.63E+14	1.91E+21	31.525	3.89182	49
50	125000	6.38E+08	5.64386	5.63E+16	50	1.13E+15	5.18E+21	31.8531	3.91202	50
51	132651	9.56E+08	5.67243	1.15E+17	51	2.25E+15	1.41E+22	32.1764	3.93183	51
52	140608	1.43E+09	5.70044	2.34E+17	52	4.50E+15	3.83E+22	32.495	3.95124	52
53	148877	2.15E+09	5.72792	4.77E+17	53	9.01E+15	1.04E+23	32.8091	3.97029	53
54	157464	3.23E+09	5.75489	9.73E+17	54	1.80E+16	2.83E+23	33.1187	3.98898	54
55	166375	4.84E+09	5.78136	1.98E+18	55	3.60E+16	7.69E+23	33.4241	4.00733	55
56	175616	7.26E+09	5.80735	4.04E+18	56	7.21E+16	2.09E+24	33.7254	4.02535	56
57	185193	1.09E+10	5.83289	8.21E+18	57	1.44E+17	5.69E+24	34.0226	4.04305	57
58	195112	1.63E+10	5.85798	1.67E+19	58	2.88E+17	1.55E+25	34.3159	4.06044	58
59	205379	2.45E+10	5.88264	3.40E+19	59	5.76E+17	4.20E+25	34.6055	4.07754	59
60	216000	3.68E+10	5.90689	6.92E+19	60	1.15E+18	1.14E+26	34.8914	4.09434	60
61	226981	5.52E+10	5.93074	1.41E+20	61	2.31E+18	3.10E+26	35.1736	4.11087	61
62	238328	8.27E+10	5.9542	2.86E+20	62	4.61E+18	8.44E+26	35.4525	4.12713	62
63	250047	1.24E+11	5.97728	5.81E+20	63	9.22E+18	2.29E+27	35.7279	4.14313	63
64	262144	1.86E+11	6	1.18E+21	64	1.84E+19	6.24E+27	36	4.15888	64
65	274625	2.79E+11	6.02237	2.40E+21	65	3.69E+19	1.69E+28	36.2689	4.17439	65
66	287496	4.19E+11	6.04439	4.87E+21	66	7.38E+19	4.61E+28	36.5347	4.18965	66
67	300763	6.28E+11	6.06609	9.89E+21	67	1.48E+20	1.25E+29	36.7974	4.20469	67
68	314432	9.42E+11	6.08746	2.01E+22	68	2.95E+20	3.40E+29	37.0572	4.21951	68
69	328509	1.41E+12	6.10852	4.07E+22	69	5.90E+20	9.25E+29	37.3141	4.23411	69
70	343000	2.12E+12	6.12928	8.26E+22	70	1.18E+21	2.52E+30	37.5681	4.2485	70
71	357911	3.18E+12	6.14975	1.68E+23	71	2.36E+21	6.84E+30	37.8194	4.26268	71
72	373248	4.77E+12	6.16993	3.40E+23	72	4.72E+21	1.86E+31	38.068	4.27667	72
73	389017	7.16E+12	6.18982	6.89E+23	73	9.44E+21	5.05E+31	38.3139	4.29046	73
74	405224	1.07E+13	6.20945	1.40E+24	74	1.89E+22	1.37E+32	38.5573	4.30407	74
75	421875	1.61E+13	6.22882	2.83E+24	75	3.78E+22	3.73E+32	38.7982	4.31749	75
76	438976	2.42E+13	6.24793	5.74E+24	76	7.56E+22	1.01E+33	39.0366	4.33073	76
77	456533	3.62E+13	6.26679	1.16E+25	77	1.51E+23	2.76E+33	39.2726	4.34381	77
78	474552	5.43E+13	6.2854	2.36E+25	78	3.02E+23	7.50E+33	39.5063	4.35671	78
79	493039	8.15E+13	6.30378	4.78E+25	79	6.04E+23	2.04E+34	39.7377	4.36945	79
80	512000	1.22E+14	6.32193	9.67E+25	80	1.21E+24	5.54E+34	39.9668	4.38203	80
81	531441	1.83E+14	6.33985	1.96E+26	81	2.42E+24	1.51E+35	40.1937	4.39445	81
82	551368	2.75E+14	6.35755	3.97E+26	82	4.84E+24	4.09E+35	40.4185	4.40672	82
83	571787	4.13E+14	6.37504	8.03E+26	83	9.67E+24	1.11E+36	40.6411	4.41884	83
84	592704	6.19E+14	6.39232	1.62E+27	84	1.93E+25	3.03E+36	40.8617	4.43082	84
85	614125	9.28E+14	6.40939	3.29E+27	85	3.87E+25	8.22E+36	41.0803	4.44265	85
86	636056	1.39E+15	6.42626	6.65E+27	86	7.74E+25	2.24E+37	41.2969	4.45435	86
87	658503	2.09E+15	6.44294	1.35E+28	87	1.55E+26	6.08E+37	41.5115	4.46591	87
88	681472	3.13E+15	6.45943	2.72E+28	88	3.09E+26	1.65E+38	41.7243	4.47734	88

89	704969	4.70E+15	6.47573	5.51E+28	89	6.19E+26	4.49E+38	41.9351	4.48864	89
90	729000	7.05E+15	6.49185	1.11E+29	90	1.24E+27	1.22E+39	42.1442	4.49981	90
91	753571	1.06E+16	6.50779	2.25E+29	91	2.48E+27	3.32E+39	42.3514	4.51086	91
92	778688	1.59E+16	6.52356	4.56E+29	92	4.95E+27	9.02E+39	42.5569	4.52179	92
93	804357	2.38E+16	6.53916	9.21E+29	93	9.90E+27	2.45E+40	42.7606	4.5326	93
94	830584	3.57E+16	6.55459	1.86E+30	94	1.98E+28	6.66E+40	42.9626	4.54329	94
95	857375	5.35E+16	6.56986	3.76E+30	95	3.96E+28	1.81E+41	43.163	4.55388	95
96	884736	8.03E+16	6.58496	7.61E+30	96	7.92E+28	4.92E+41	43.3617	4.56435	96
97	912673	1.20E+17	6.59991	1.54E+31	97	1.58E+29	1.34E+42	43.5588	4.57471	97
98	941192	1.81E+17	6.61471	3.11E+31	98	3.17E+29	3.64E+42	43.7544	4.58497	98
99	970299	2.71E+17	6.62936	6.27E+31	99	6.34E+29	9.89E+42	43.9484	4.59512	99
100	1.00E+06	4.07E+17	6.64386	1.27E+32	100	1.27E+30	2.69E+43	44.1408	4.60517	100

n! :

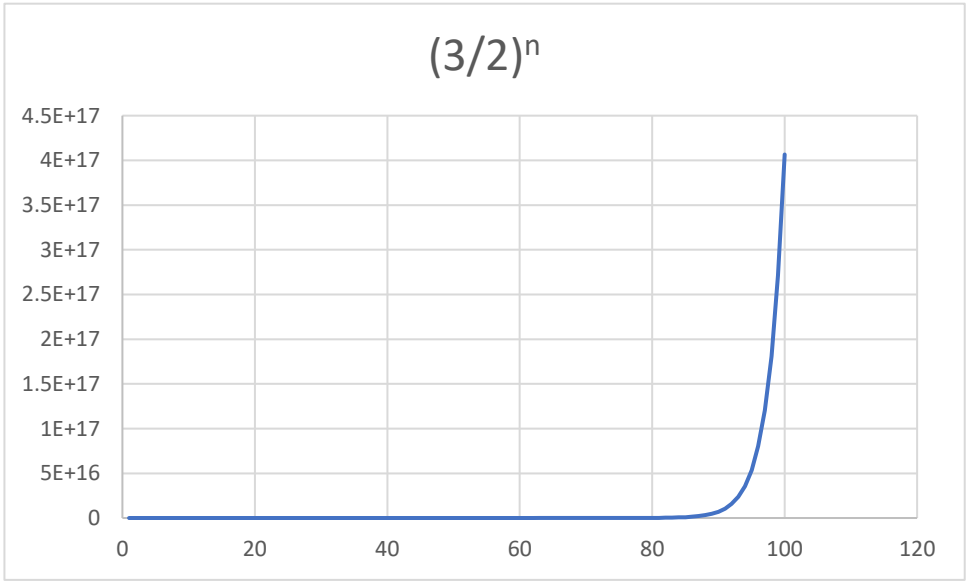
1-1
2-2
3-6
4-24
5-120
6-720
7-5040
8-40320
9-362880
10-3.6288e+006
11-3.99168e+007
12-4.79002e+008
13-6.22702e+009
14-8.71783e+010
15-1.30767e+012
16-2.09228e+013
17-3.55687e+014
18-6.40237e+015
19-1.21645e+017
20-2.4329e+018

Graph:



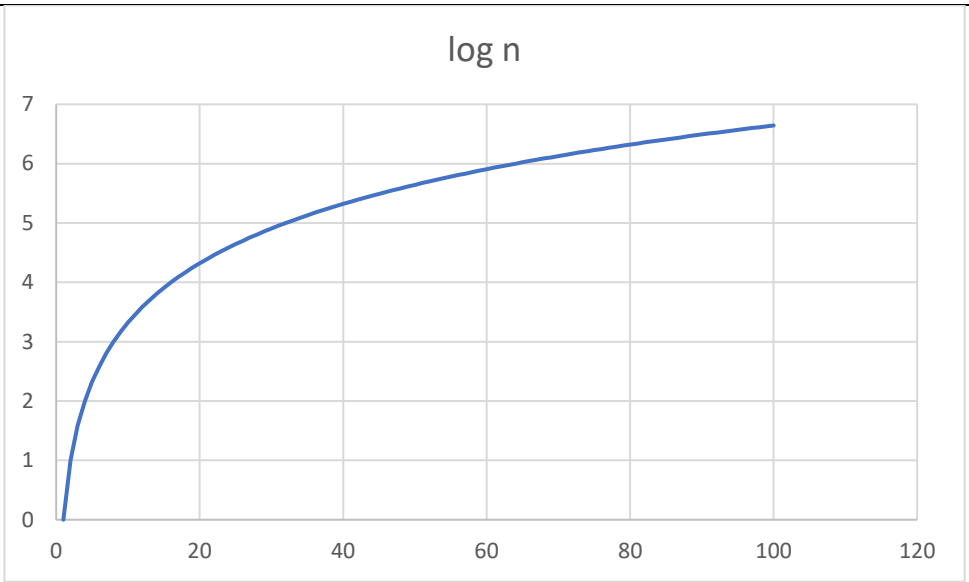
- **Inference**:

Here we can notice curvature increase in value of (n^3) from 0 to 100000(Value of N from 0 to 100). This growth shows us that value of (N) grows exponentially as we consider higher (N) Value



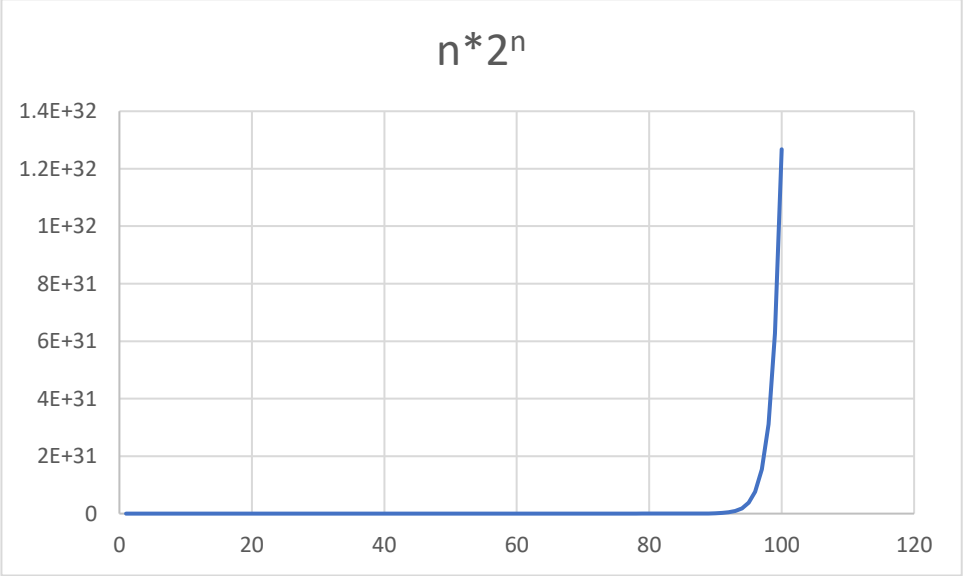
- **Inference**:

Here we can see that there is no cosiderate growth in (3/2)^n value until the value on n=90. After that we can see an upright growth in value till value of n=100.



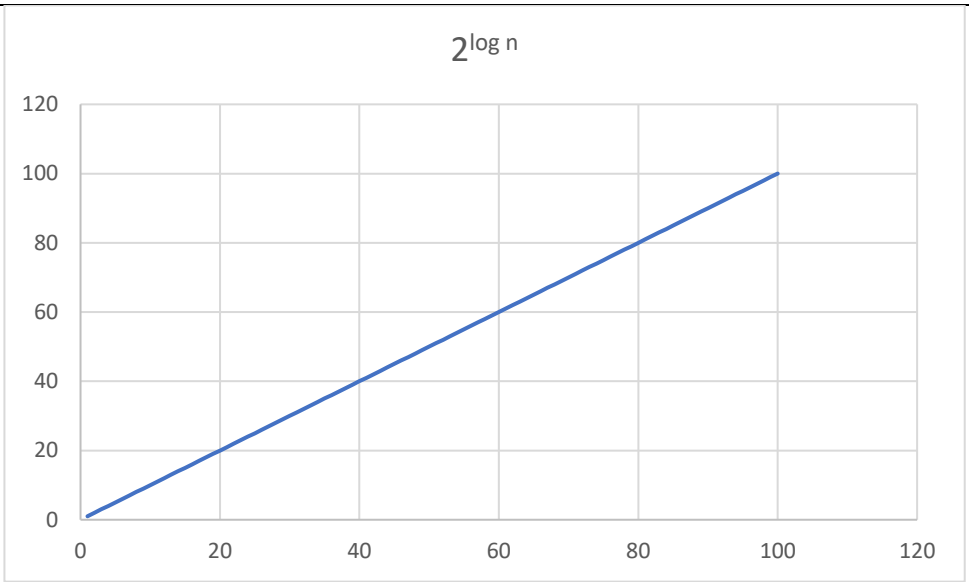
- **Inference**:

Here first there is an upright growth for value of logn until value of n=10. After then that the proportinal growth decreases till value of n is 100.



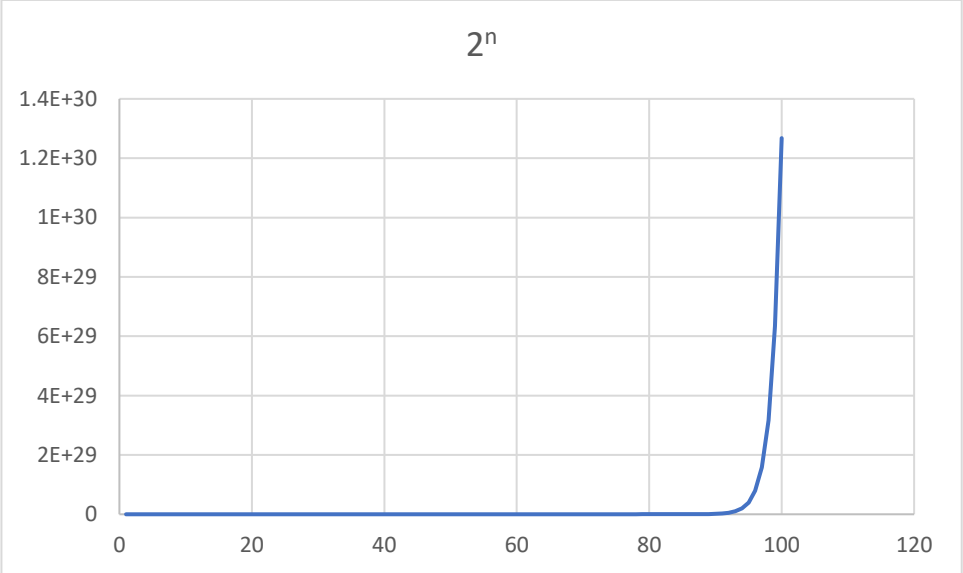
- **Inference**:

Here we can see that there is no cosiderate growth in (n*2)^n value until the value on n=95. After that we can see an upright growth in value till value of n=100.



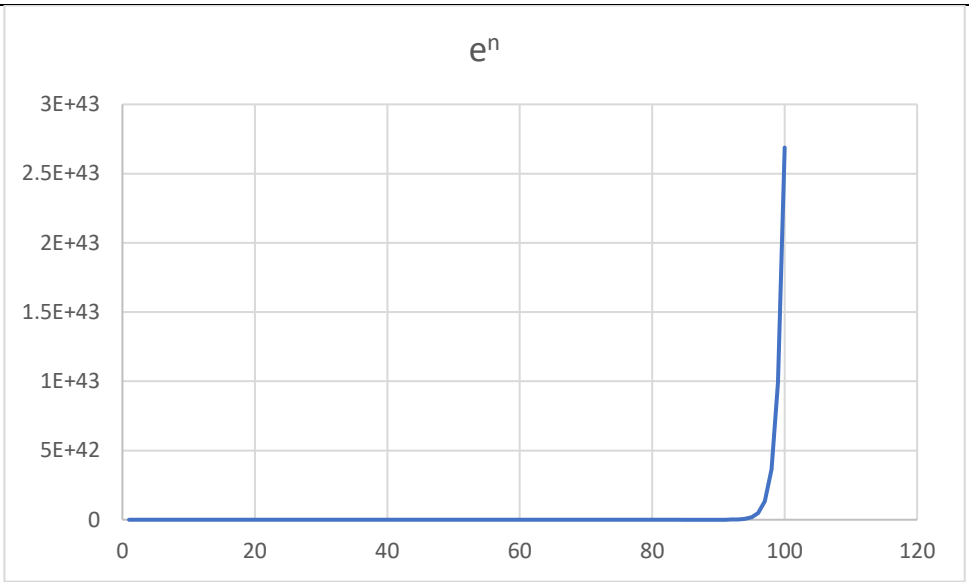
- **Inference**:

Here we can see that there is a constant growth for every value of n from 1 to 100



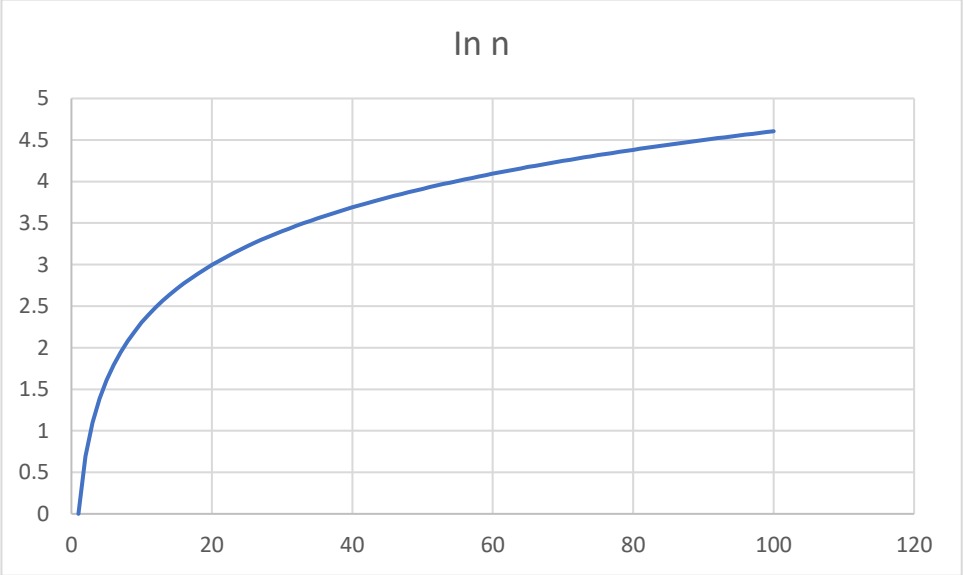
- **Inference**:

Here we can see that there is no cosiderate growth in (2)^n value until the value on n=95. After that we can see an upright growth in value till value of n=100.



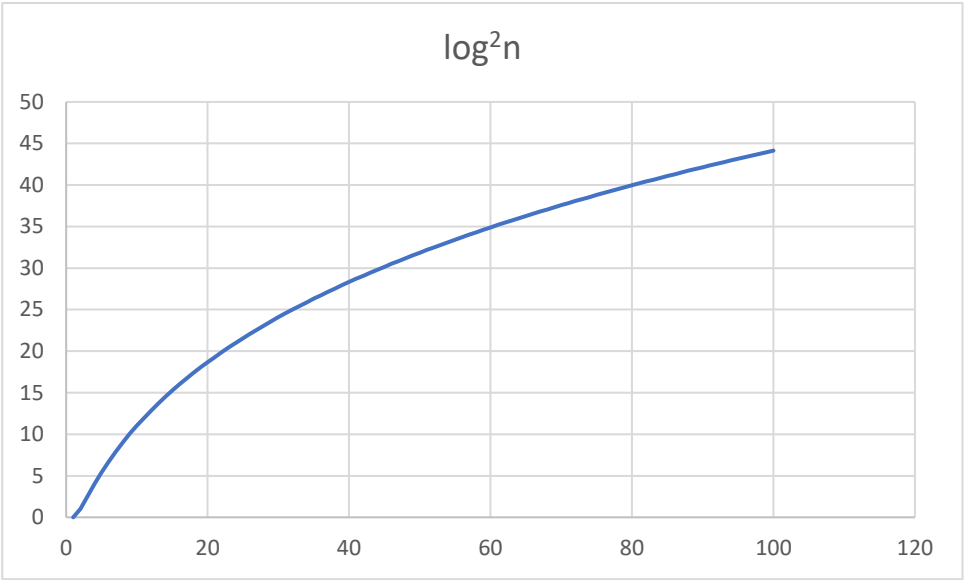
- **Inference**:

Here we can see that there is no cosiderate growth in (e^n) value until the value on n=95. After that we can see an upright growth in value till value of n=100.



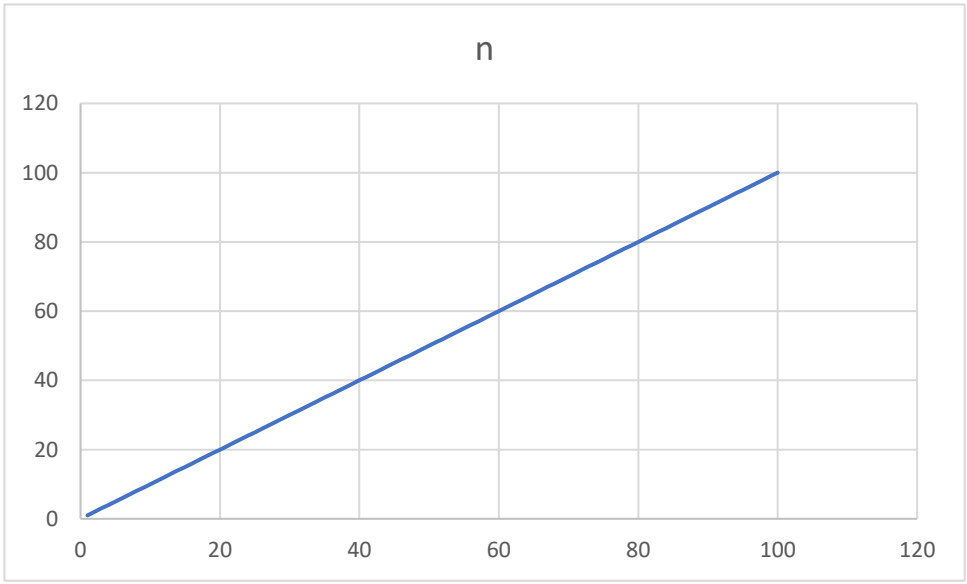
- **Inference**:

Here first there is an upright growth for value of (ln n) until value of n=10. After then that the proportinal growth decreases till value of n is 100.



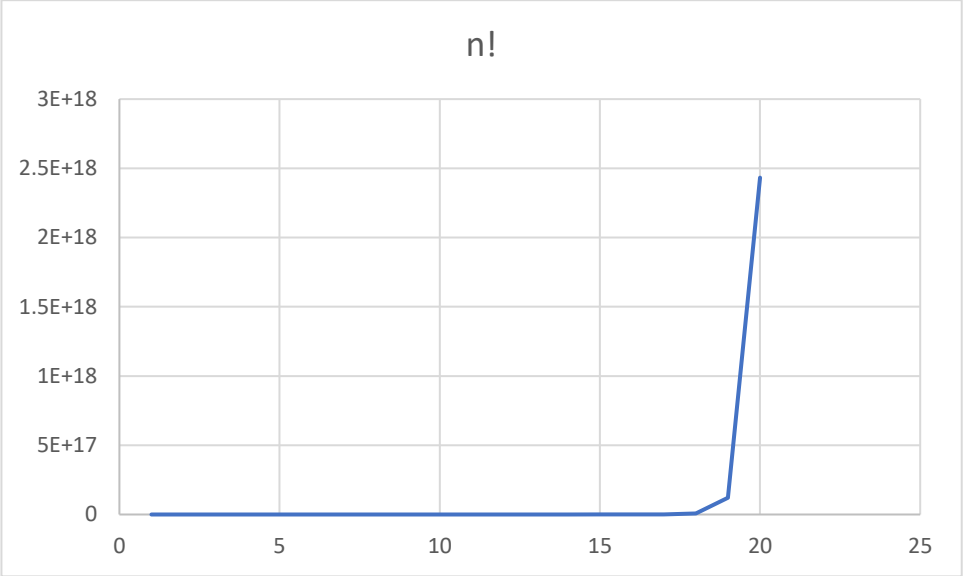
- **Inference**:

Here first there is an curvature upright growth for value of (log^2)n for every value of n.



- **Inference**:

Here we can see that there is a constant growth for every value of n from 1 to 100.



- **Inference**:

Here we can see that there is no cosiderate growth in (e^n) value until the value on n=18. After that we can see an upright growth in value till value of n=20.

Conclusion:	Through this experiment, I gained a comprehensive understanding of utilization logarithmic and exponential functions in C programming language and the implementation of recursive functions, enhancing my programming skills and knowledge.