NAME:	Vinit Madhyan
UID:	2021300070
SUBJECT	Design and Analysis of Algorithm
EXPERIMENT NO :	1
AIM:	To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.
	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 thevalues for result. Function 3: i. Initialize variables n and result. ii. result = 3n/2 iii. Apply a for loop for values of n from 0-100 and print all thevalues 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 thevalues for result. Function 5: 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

	thevalues for result. Function 6: i. Initialize variables n and result. ii. result = pow(2,n) iii. Apply a for loop for values of n from 0-100 and print all thevalues for result. Function 7: i. Initialize variables n and result. ii. result = pow(e,n) iii. Apply a for loop for values of n from 0-100 and print all thevalues for result. Function 8: i. Initialize variables n and result. ii. result = pow(2,log10(n)) iii. Apply a for loop for values of n from 0-100 and print all thevalues for result. Function 9: iv. Initialize variables n and result. v. result = n*pow(2,n) vi. Apply a for loop for values of n from 0-100 and print all thevalues for result. Function 10: i. Initialize variables n and result. ii. result = sqrt(log10(n)) iii. Apply a for loop for values of n from 0-100 and print all thevalues for result. Function 11: i. Initialize variables n and result. ii. result = pow(sqrt(2),log10(n)) iii. Apply a for loop for values of n from 0-100 and print all thevalues for result.

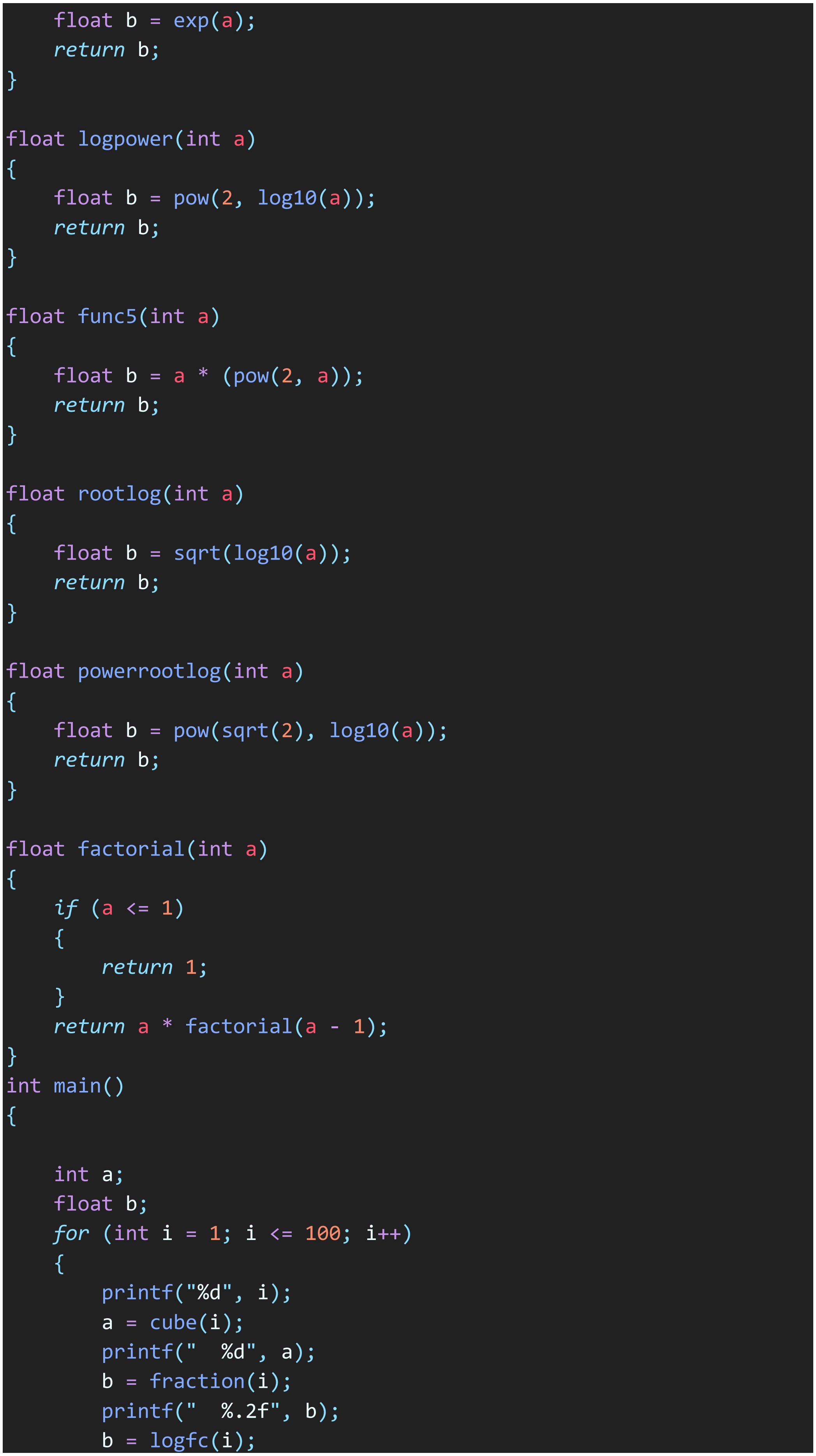
Function 12:

```
i.    Initialize a variable n
ii. Create a function to find the factorial.
iii. factorial(n)
        if(n==1 || n==0)
            return i
        else
            return n*factorial(n-1)
```

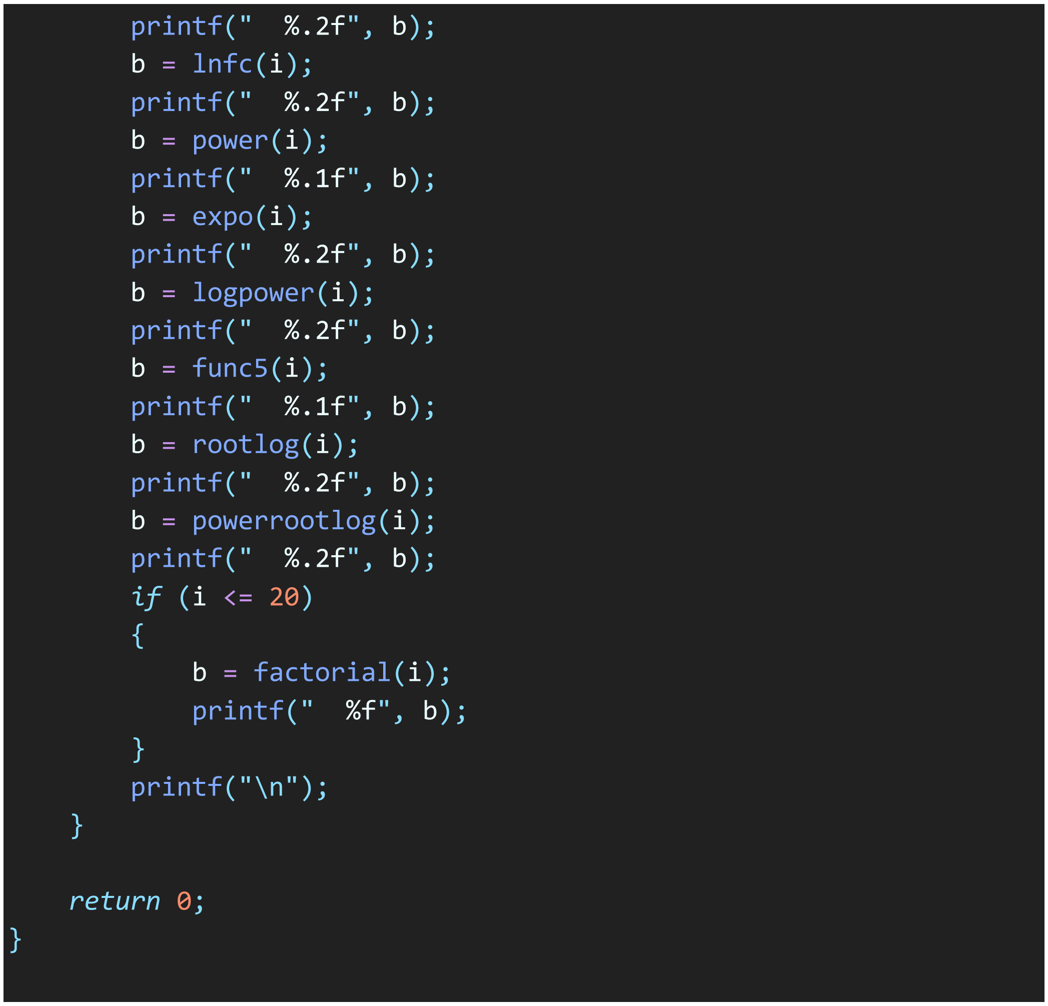
Apply a for loop for values of n from 0-19 and print al the values for result in the main function.

PROGRAM:

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

int cube(int a)
{
    return a * a * a;
}

float fraction(int a)
{
    int b = 3 * a;
    float c = (float)(b) / (float)(2);
    return c;
}

float logfc(int a)
{
    float c = log10(a);
    return c;
}
float lnfc(int a)
{
    float c = log(a);
    return c;
}

float power(int a)
{
    float b = pow(2, a);
    return b;
}

float expo(int a)
{
```

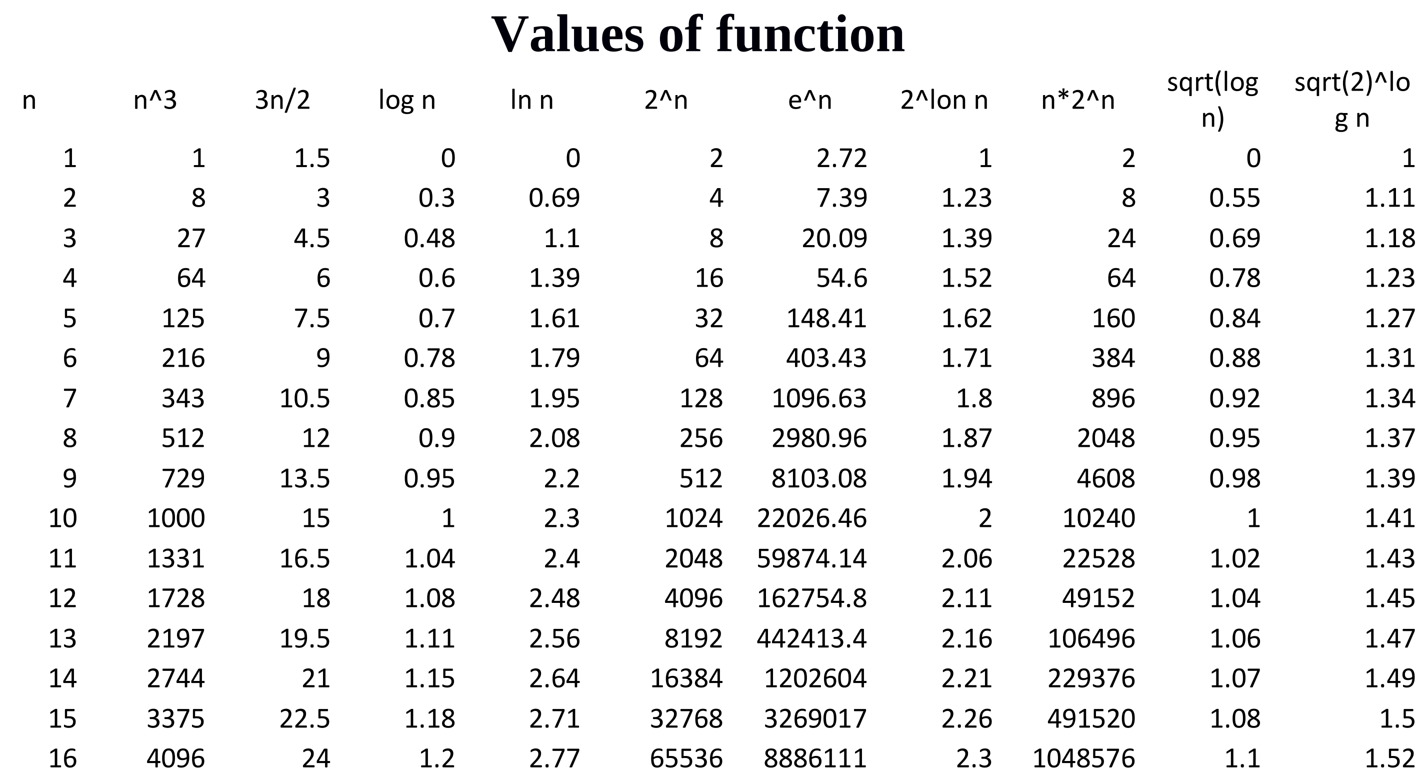


```c
    float b = exp(a);
    return b;
}

float logpower(int a)
{
    float b = pow(2, log10(a));
    return b;
}

float func5(int a)
{
    float b = a * (pow(2, a));
    return b;
}

float rootlog(int a)
{
    float b = sqrt(log10(a));
    return b;
}

float powerrootlog(int a)
{
    float b = pow(sqrt(2), log10(a));
    return b;
}

float factorial(int a)
{
    if (a <= 1)
    {
        return 1;
    }
    return a * factorial(a - 1);
}
int main()
{

    int a;
    float b;
    for (int i = 1; i <= 100; i++)
    {
        printf("%d", i);
        a = cube(i);
        printf("  %d", a);
        b = fraction(i);
        printf("  %.2f", b);
        b = logfc(i);
```



```
        printf("  %.2f", b);
        b = lnfc(i);
        printf("  %.2f", b);
        b = power(i);
        printf("  %.1f", b);
        b = expo(i);
        printf("  %.2f", b);
        b = logpower(i);
        printf("  %.2f", b);
        b = func5(i);
        printf("  %.1f", b);
        b = rootlog(i);
        printf("  %.2f", b);
        b = powerrootlog(i);
        printf("  %.2f", b);
        if (i <= 20)
        {
            b = factorial(i);
            printf("  %f", b);
        }
        printf("\n");
    }

    return 0;
}
```

Observation (SNAPSHOT)



Values of function

n	n^3	3n/2	log n	ln n	2^n	e^n	2^lon n	n*2^n	sqrt(log n)	sqrt(2)^log n
1	1	1.5	0	0	2	2.72	1	2	0	1
2	8	3	0.3	0.69	4	7.39	1.23	8	0.55	1.11
3	27	4.5	0.48	1.1	8	20.09	1.39	24	0.69	1.18
4	64	6	0.6	1.39	16	54.6	1.52	64	0.78	1.23
5	125	7.5	0.7	1.61	32	148.41	1.62	160	0.84	1.27
6	216	9	0.78	1.79	64	403.43	1.71	384	0.88	1.31
7	343	10.5	0.85	1.95	128	1096.63	1.8	896	0.92	1.34
8	512	12	0.9	2.08	256	2980.96	1.87	2048	0.95	1.37
9	729	13.5	0.95	2.2	512	8103.08	1.94	4608	0.98	1.39
10	1000	15	1	2.3	1024	22026.46	2	10240	1	1.41
11	1331	16.5	1.04	2.4	2048	59874.14	2.06	22528	1.02	1.43
12	1728	18	1.08	2.48	4096	162754.8	2.11	49152	1.04	1.45
13	2197	19.5	1.11	2.56	8192	442413.4	2.16	106496	1.06	1.47
14	2744	21	1.15	2.64	16384	1202604	2.21	229376	1.07	1.49
15	3375	22.5	1.18	2.71	32768	3269017	2.26	491520	1.08	1.5
16	4096	24	1.2	2.77	65536	8886111	2.3	1048576	1.1	1.52

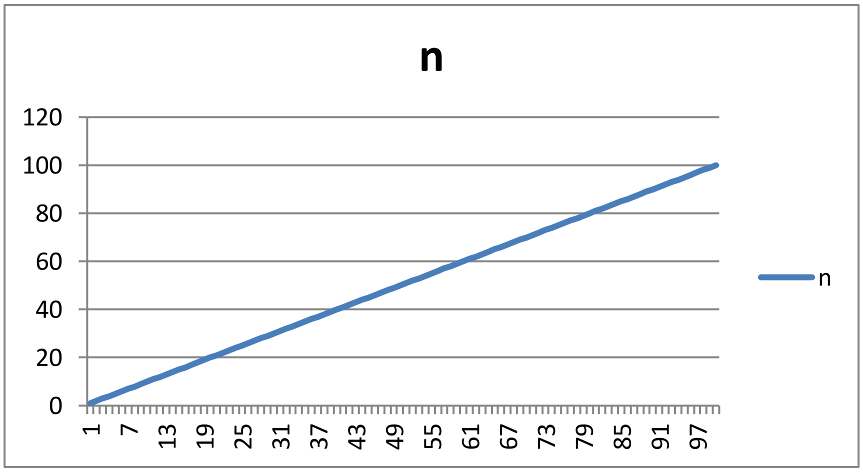
17	4913	25.5	1.23	2.83	131072	24154952	2.35	2228224	1.11	1.53
18	5832	27	1.26	2.89	262144	65659968	2.39	4718592	1.12	1.55
19	6859	28.5	1.28	2.94	524288	1.78E+08	2.43	9961472	1.13	1.56
20	8000	30	1.3	3	1048576	4.85E+08	2.46	20971520	1.14	1.57
21	9261	31.5	1.32	3.04	2097152	1.32E+09	2.5	44040192	1.15	1.58
22	10648	33	1.34	3.09	4194304	3.58E+09	2.54	92274688	1.16	1.59
23	12167	34.5	1.36	3.14	8388608	9.74E+09	2.57	1.93E+08	1.17	1.6
24	13824	36	1.38	3.18	16777216	2.65E+10	2.6	4.03E+08	1.17	1.61
25	15625	37.5	1.4	3.22	33554432	7.2E+10	2.64	8.39E+08	1.18	1.62
26	17576	39	1.41	3.26	67108864	1.96E+11	2.67	1.74E+09	1.19	1.63
27	19683	40.5	1.43	3.3	1.34E+08	5.32E+11	2.7	3.62E+09	1.2	1.64
28	21952	42	1.45	3.33	2.68E+08	1.45E+12	2.73	7.52E+09	1.2	1.65
29	24389	43.5	1.46	3.37	5.37E+08	3.93E+12	2.76	1.56E+10	1.21	1.66
30	27000	45	1.48	3.4	1.07E+09	1.07E+13	2.78	3.22E+10	1.22	1.67
31	29791	46.5	1.49	3.43	2.15E+09	2.9E+13	2.81	6.66E+10	1.22	1.68
32	32768	48	1.51	3.47	4.29E+09	7.9E+13	2.84	1.37E+11	1.23	1.68
33	35937	49.5	1.52	3.5	8.59E+09	2.15E+14	2.86	2.83E+11	1.23	1.69
34	39304	51	1.53	3.53	1.72E+10	5.83E+14	2.89	5.84E+11	1.24	1.7
35	42875	52.5	1.54	3.56	3.44E+10	1.59E+15	2.92	1.2E+12	1.24	1.71
36	46656	54	1.56	3.58	6.87E+10	4.31E+15	2.94	2.47E+12	1.25	1.71
37	50653	55.5	1.57	3.61	1.37E+11	1.17E+16	2.97	5.09E+12	1.25	1.72
38	54872	57	1.58	3.64	2.75E+11	3.19E+16	2.99	1.04E+13	1.26	1.73
39	59319	58.5	1.59	3.66	5.5E+11	8.66E+16	3.01	2.14E+13	1.26	1.74
40	64000	60	1.6	3.69	1.1E+12	2.35E+17	3.04	4.4E+13	1.27	1.74
41	68921	61.5	1.61	3.71	2.2E+12	6.4E+17	3.06	9.02E+13	1.27	1.75
42	74088	63	1.62	3.74	4.4E+12	1.74E+18	3.08	1.85E+14	1.27	1.76
43	79507	64.5	1.63	3.76	8.8E+12	4.73E+18	3.1	3.78E+14	1.28	1.76
44	85184	66	1.64	3.78	1.76E+13	1.29E+19	3.12	7.74E+14	1.28	1.77
45	91125	67.5	1.65	3.81	3.52E+13	3.49E+19	3.15	1.58E+15	1.29	1.77
46	97336	69	1.66	3.83	7.04E+13	9.5E+19	3.17	3.24E+15	1.29	1.78
47	103823	70.5	1.67	3.85	1.41E+14	2.58E+20	3.19	6.61E+15	1.29	1.79
48	110592	72	1.68	3.87	2.81E+14	7.02E+20	3.21	1.35E+16	1.3	1.79
49	117649	73.5	1.69	3.89	5.63E+14	1.91E+21	3.23	2.76E+16	1.3	1.8
50	125000	75	1.7	3.91	1.13E+15	5.18E+21	3.25	5.63E+16	1.3	1.8
51	132651	76.5	1.71	3.93	2.25E+15	1.41E+22	3.27	1.15E+17	1.31	1.81
52	140608	78	1.72	3.95	4.5E+15	3.83E+22	3.29	2.34E+17	1.31	1.81
53	148877	79.5	1.72	3.97	9.01E+15	1.04E+23	3.3	4.77E+17	1.31	1.82
54	157464	81	1.73	3.99	1.8E+16	2.83E+23	3.32	9.73E+17	1.32	1.82
55	166375	82.5	1.74	4.01	3.6E+16	7.69E+23	3.34	1.98E+18	1.32	1.83
56	175616	84	1.75	4.03	7.21E+16	2.09E+24	3.36	4.04E+18	1.32	1.83

57	185193	85.5	1.76	4.04	1.44E+17	5.69E+24	3.38	8.21E+18	1.33	1.84
58	195112	87	1.76	4.06	2.88E+17	1.55E+25	3.4	1.67E+19	1.33	1.84
59	205379	88.5	1.77	4.08	5.76E+17	4.2E+25	3.41	3.4E+19	1.33	1.85
60	216000	90	1.78	4.09	1.15E+18	1.14E+26	3.43	6.92E+19	1.33	1.85
61	226981	91.5	1.79	4.11	2.31E+18	3.1E+26	3.45	1.41E+20	1.34	1.86
62	238328	93	1.79	4.13	4.61E+18	8.44E+26	3.46	2.86E+20	1.34	1.86
63	250047	94.5	1.8	4.14	9.22E+18	2.29E+27	3.48	5.81E+20	1.34	1.87
64	262144	96	1.81	4.16	1.84E+19	6.24E+27	3.5	1.18E+21	1.34	1.87
65	274625	97.5	1.81	4.17	3.69E+19	1.69E+28	3.51	2.4E+21	1.35	1.87
66	287496	99	1.82	4.19	7.38E+19	4.61E+28	3.53	4.87E+21	1.35	1.88
67	300763	100.5	1.83	4.2	1.48E+20	1.25E+29	3.55	9.89E+21	1.35	1.88
68	314432	102	1.83	4.22	2.95E+20	3.4E+29	3.56	2.01E+22	1.35	1.89
69	328509	103.5	1.84	4.23	5.9E+20	9.25E+29	3.58	4.07E+22	1.36	1.89
70	343000	105	1.85	4.25	1.18E+21	2.52E+30	3.59	8.26E+22	1.36	1.9
71	357911	106.5	1.85	4.26	2.36E+21	6.84E+30	3.61	1.68E+23	1.36	1.9
72	373248	108	1.86	4.28	4.72E+21	1.86E+31	3.62	3.4E+23	1.36	1.9
73	389017	109.5	1.86	4.29	9.44E+21	5.05E+31	3.64	6.89E+23	1.37	1.91
74	405224	111	1.87	4.3	1.89E+22	1.37E+32	3.65	1.4E+24	1.37	1.91
75	421875	112.5	1.88	4.32	3.78E+22	3.73E+32	3.67	2.83E+24	1.37	1.92
76	438976	114	1.88	4.33	7.56E+22	1.01E+33	3.68	5.74E+24	1.37	1.92
77	456533	115.5	1.89	4.34	1.51E+23	2.76E+33	3.7	1.16E+25	1.37	1.92
78	474552	117	1.89	4.36	3.02E+23	7.5E+33	3.71	2.36E+25	1.38	1.93
79	493039	118.5	1.9	4.37	6.04E+23	2.04E+34	3.73	4.78E+25	1.38	1.93
80	512000	120	1.9	4.38	1.21E+24	5.54E+34	3.74	9.67E+25	1.38	1.93
81	531441	121.5	1.91	4.39	2.42E+24	1.51E+35	3.75	1.96E+26	1.38	1.94
82	551368	123	1.91	4.41	4.84E+24	4.09E+35	3.77	3.97E+26	1.38	1.94
83	571787	124.5	1.92	4.42	9.67E+24	1.11E+36	3.78	8.03E+26	1.39	1.94
84	592704	126	1.92	4.43	1.93E+25	3.03E+36	3.8	1.62E+27	1.39	1.95
85	614125	127.5	1.93	4.44	3.87E+25	8.22E+36	3.81	3.29E+27	1.39	1.95
86	636056	129	1.93	4.45	7.74E+25	2.24E+37	3.82	6.65E+27	1.39	1.96
87	658503	130.5	1.94	4.47	1.55E+26	6.08E+37	3.84	1.35E+28	1.39	1.96
88	681472	132	1.94	4.48	3.09E+26	1.65E+38	3.85	2.72E+28	1.39	1.96
89	704969	133.5	1.95	4.49	6.19E+26	1.#J	3.86	5.51E+28	1.4	1.97
90	729000	135	1.95	4.5	1.24E+27	1.#J	3.88	1.11E+29	1.4	1.97
91	753571	136.5	1.96	4.51	2.48E+27	1.#J	3.89	2.25E+29	1.4	1.97
92	778688	138	1.96	4.52	4.95E+27	1.#J	3.9	4.56E+29	1.4	1.98
93	804357	139.5	1.97	4.53	9.9E+27	1.#J	3.91	9.21E+29	1.4	1.98
94	830584	141	1.97	4.54	1.98E+28	1.#J	3.93	1.86E+30	1.4	1.98
95	857375	142.5	1.98	4.55	3.96E+28	1.#J	3.94	3.76E+30	1.41	1.98
96	884736	144	1.98	4.56	7.92E+28	1.#J	3.95	7.61E+30	1.41	1.99
97	912673	145.5	1.99	4.57	1.58E+29	1.#J	3.96	1.54E+31	1.41	1.99
98	941192	147	1.99	4.58	3.17E+29	1.#J	3.98	3.11E+31	1.41	1.99
99	970299	148.5	2	4.6	6.34E+29	1.#J	3.99	6.27E+31	1.41	2
100	1000000	150	2	4.61	1.27E+30	1.#J	4	1.27E+32	1.41	2

Factorial

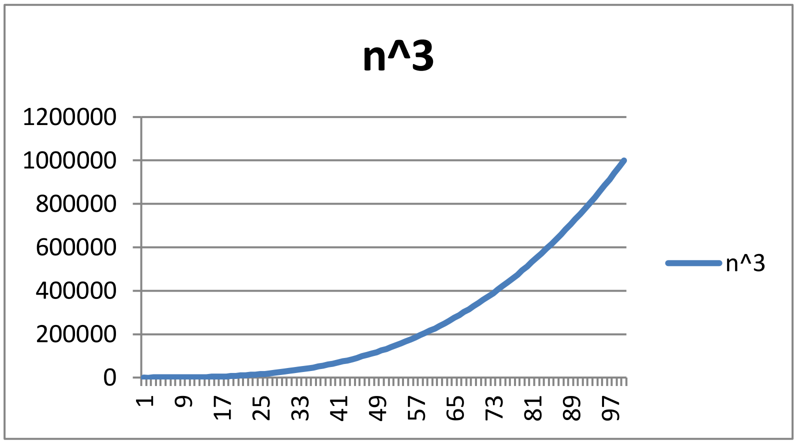
1-1
2-2
3-6
4-24
5-120
6-720
7-5040
8-40320
9-362880
10-3628800
11-39916800
12-479001600
13-6227020800
14-87178289152
15-1307674411008
16-20922790576128
17-355687414628352
18-6402373530419200
19-121645104594157570
20-2432902023163674600

Graphs



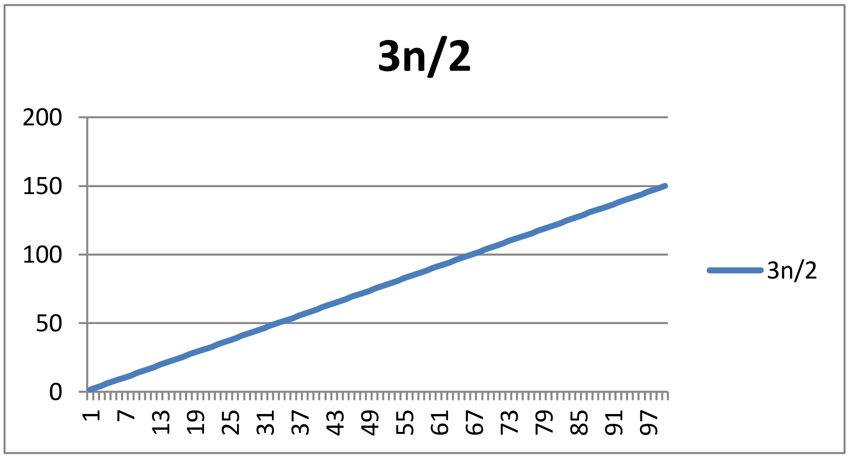
Observation

The graph of f(n)=n is a straight line with a slope of 1 and y-intercept of 0. It passes through the origin (0,0) and for every increase in n by 1 unit, there is a corresponding increase in f(n) by 1 unit.



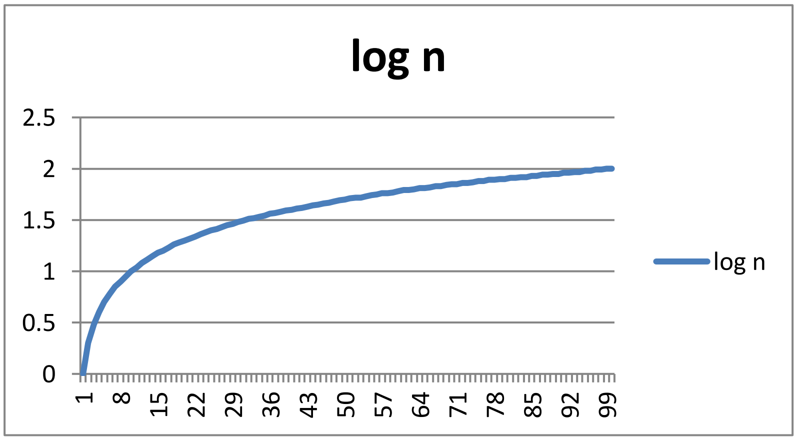
Observation

The graph of f(n)=n^3 is a smooth and continuous curve that rapidly increases as n increases from negative to positive values. The graph passes through the point (0, 0)



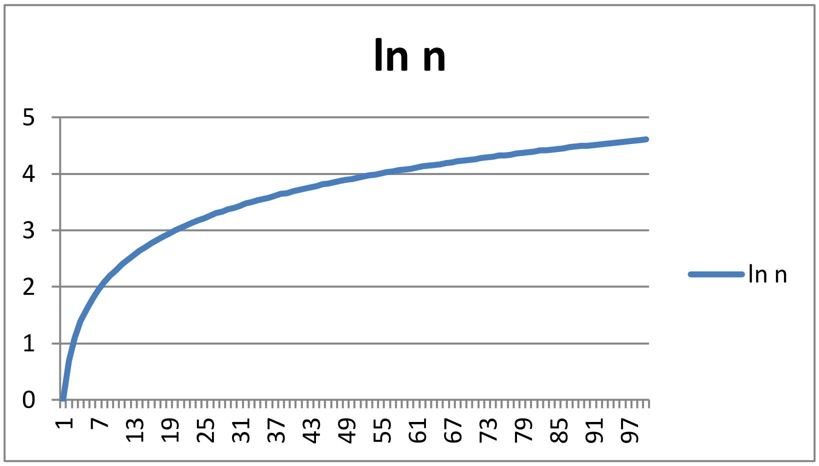
Observation

The graph of f(n)=3n/2 is a straight line with a slope of 3/2 and y-intercept of 0. It passes through the origin (0,0) .



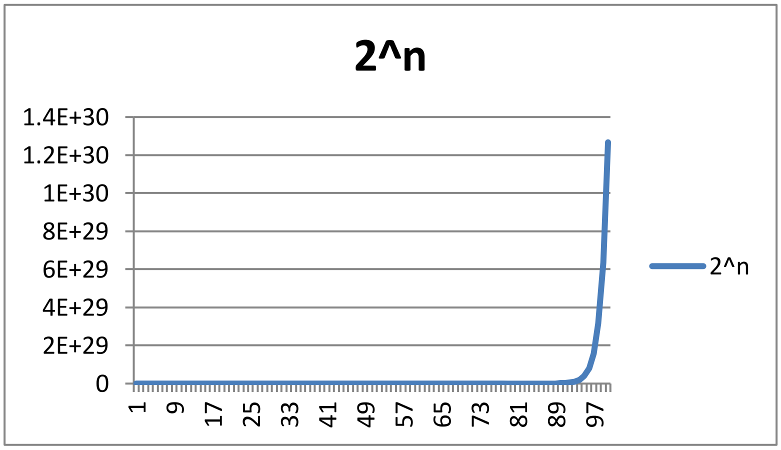
Observation

The graph of f(n)=log10(n), where log10 is the logarithm with base 10, is a smooth and continuous curve that starts at negative infinity and approaches 0 as n approaches 1. As n increases, the value of f(n) also increases but at a slower rate. The graph passes through the point (1, 0) and for every increase in n by a factor of 10, f(n) increases by 1 unit



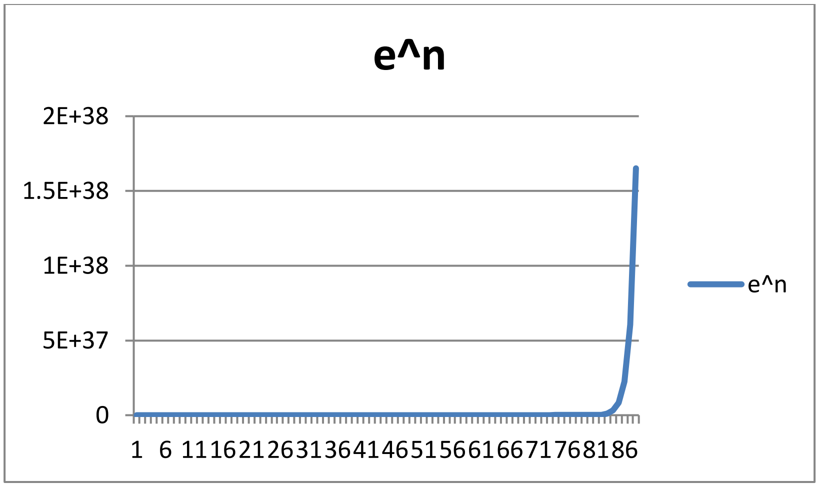
Observation

The graph of f(n)=log(n), where log is the natural logarithm (base e), is a smooth and continuous curve that starts at negative infinity and approaches 0 as n approaches 1. As n increases, the value of f(n) also increases but at a slower rate. The graph passes through the point (1, 0) and for every increase in n by a factor of e, f(n) increases by 1 unit



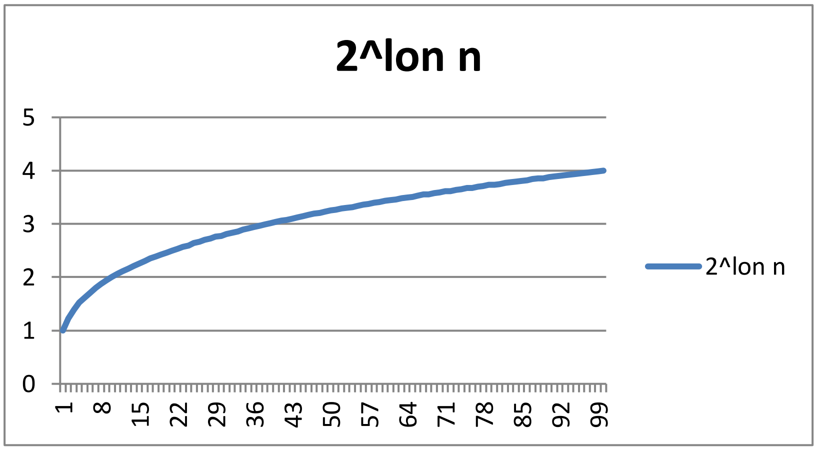
Observation

The graph of f(n)=2^n is a smooth and continuous curve that rapidly increases as n increases from negative to positive values. The graph passes through the point (0, 1)



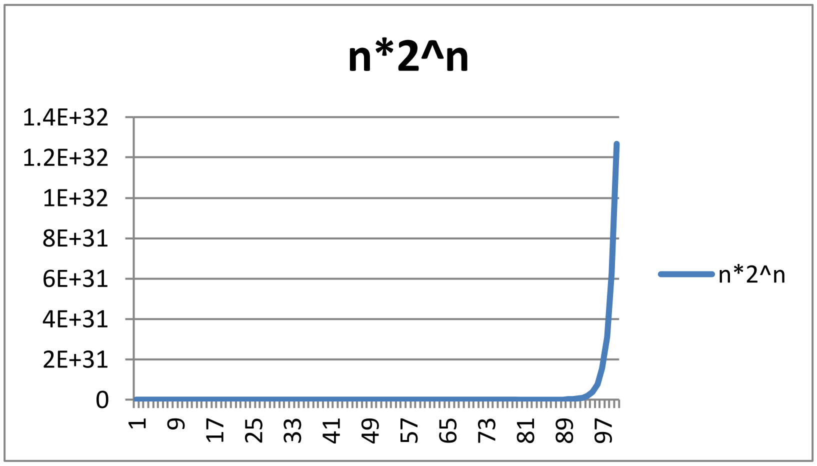
Observation

The graph of f(n)=e^n, where e is the mathematical constant approximately equal to 2.71828, is a smooth and continuous curve that starts at 0 and rapidly increases as n increases from negative to positive values. The graph passes through the point (0, 1)



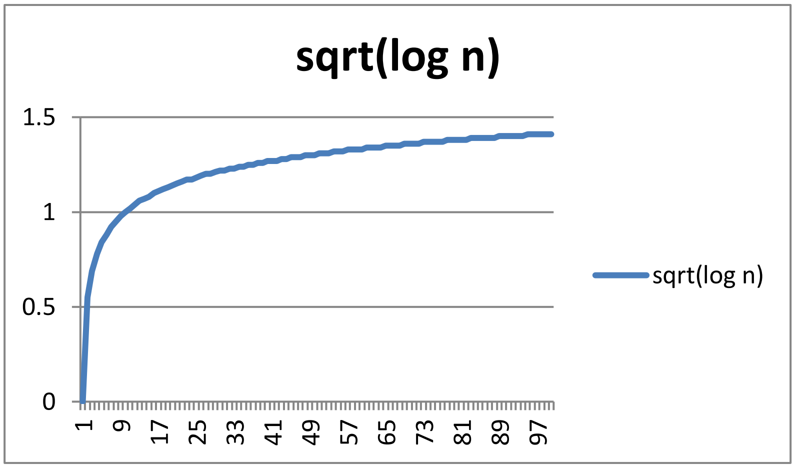
Observation

The graph of f(n)=2^(log10 n) is a smooth and continuous curve that starts at 1 and rapidly increases as n increases from 1 to positive values.



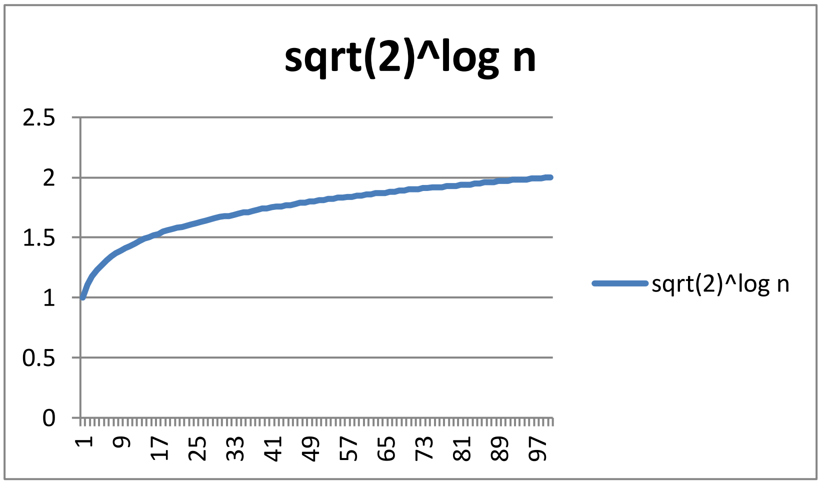
Observation

The graph of f(n)=n*2^n is a smooth and continuous curve that rapidly increases as n increases from negative to positive values. The graph passes through the point (0, 0)



Observation

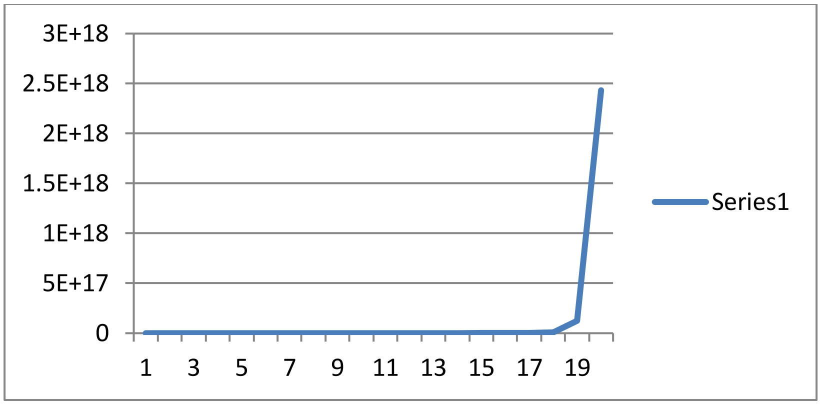
The graph of f(n)=sqrt(log10 n), where sqrt represents the square root function and log10 is the logarithm with base 10, is a smooth and continuous curve that starts at 0 and increases as n increases from 1 to positive values



Observation

The graph of f(n)=sqrt(2)^(log10 n) is a smooth and continuous curve that starts at 1 and rapidly increases as n increases from 1 to positive values.

Graph of n!



Observation

The graph of f(n)=n!, where n! represents the factorial of n (the product of all positive integers up to n), is a smooth and continuous curve that starts at 1 and rapidly increases as n increases from 0 to positive values. The graph passes through the point (0, 1) and as n approaches positive infinity, f(n) also approaches positive infinity

Conclusion

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