Experiment No.	1 A
Aim	To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.
Name	Rucha Sudhir Kulkarni
UID No.	2021300067
Class & Division	SE Computer Engineering (Div: A)(Batch-D)

Aim:
To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.

Theory:
A function is a relation between a set of inputs and a set of permissible outputs with the property that each
input is related to exactly one output. Let A & B be any two non-empty sets; mapping from A to B will be a function
only when every element in set A has one end, only one image in set B.

Problem Definition & Assumptions:
For this experiment, you have to implement at least 10 functions from the given list. The input (i.e. n) to all the functions varies from 0 to 100 with increment of 1. Then add the function n! in the list and execute the same for n from 0 to 20.

Algorithm:
1. Start
2. Iterate over all numbers from zero to 100.
3. Print the value of all the 10 functions for each iteration.
4. Iterate over numbers from zero to 20.
5. Print the values of their factorial.
6. Stop

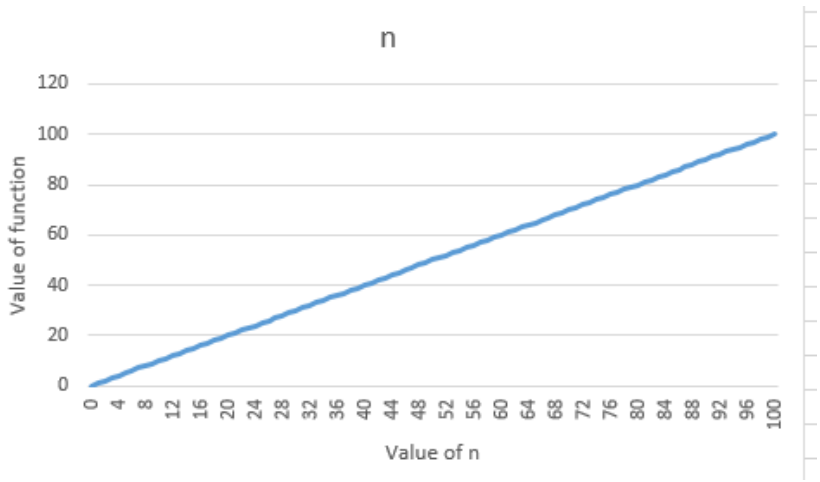
Program:

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

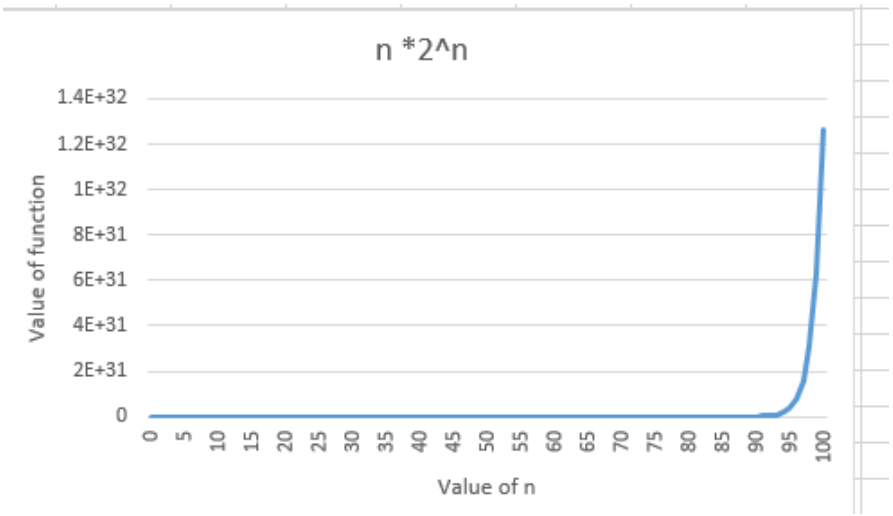
```
int ncube(int n){
return pow(n,3);
}

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

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

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

double threeby2n(int n){
return pow(1.5,n);
}

double logb2(int n){
return (log(n)/log(2));
}

float rootlogb2(int n){
return pow((log(n)/log(2)),0.5);
}

float nlogb2(int n){
return n*(log(n)/log(2));
}

double sqlogb2(int n){
return (log(n)/log(2))*(log(n)/log(2));
}

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

long long int factorial(int n){
if(n<=1)
return 1;
```

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




int main(){
for(int n=0;n<=100;n++){

printf("%d\n",ncube(n));
 printf("%lf\n",tworn(n));
 printf("%lf\n",ntworn(n));
 printf("%lf\n",threeby2n(n));
 printf("%lf\n",logb2(n));
 printf("%f\n",rootlogb2(n));
 printf("%f\n",nlogb2(n));
 printf("%lf\n",sqlogb2(n));
 printf("%lf\n",lnln(n));
 printf("%d\n",n);
}
for(int i=0;i<=20;i++){
printf("%lld\n",factorial(i));
}
return 0;
}
```

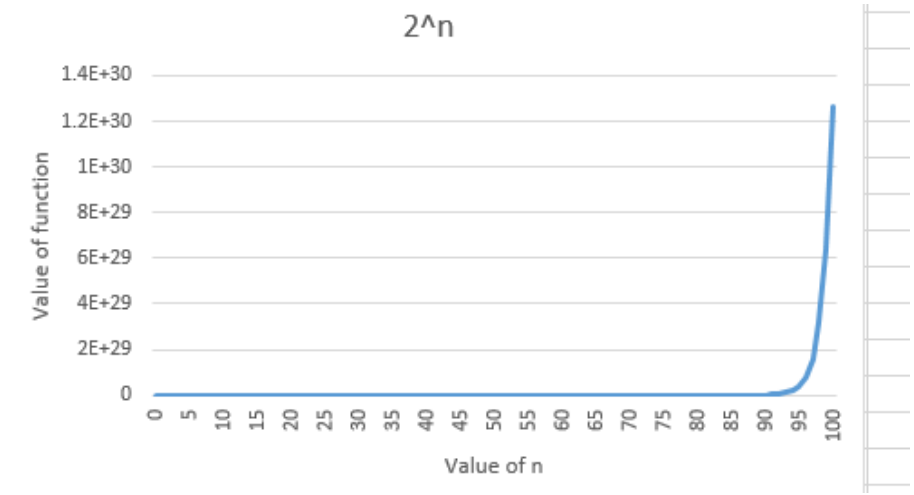
Observation (2D plot of the function values):



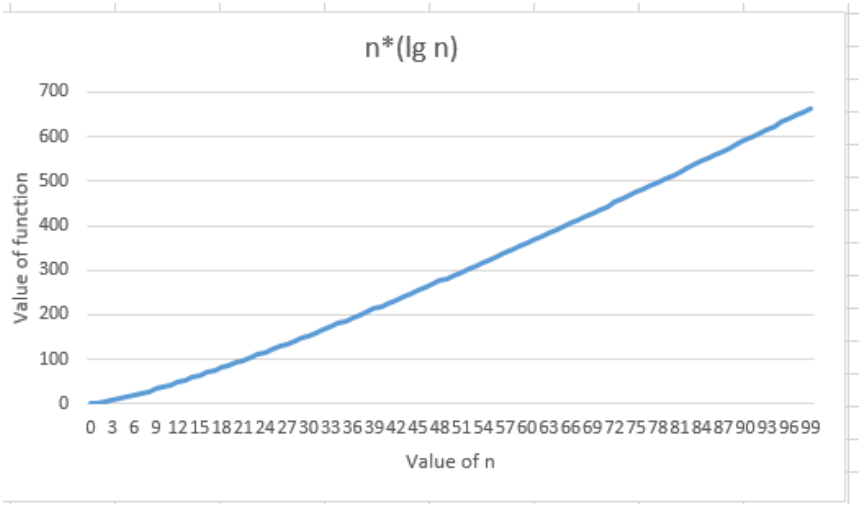
The value of function increases as the value of n increases.



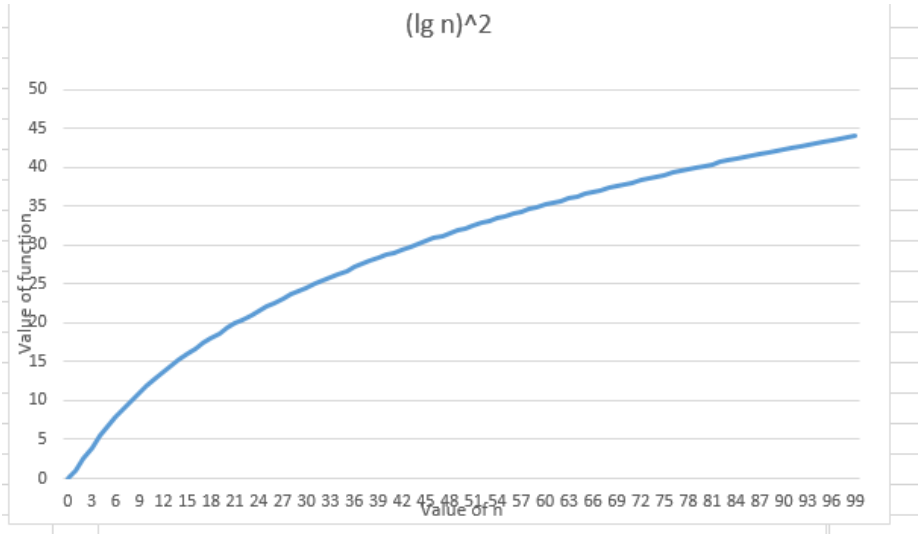
The graph is initially straight line indicating not much change in values then the values of function increases first slowly and then suddenly shoots up high.



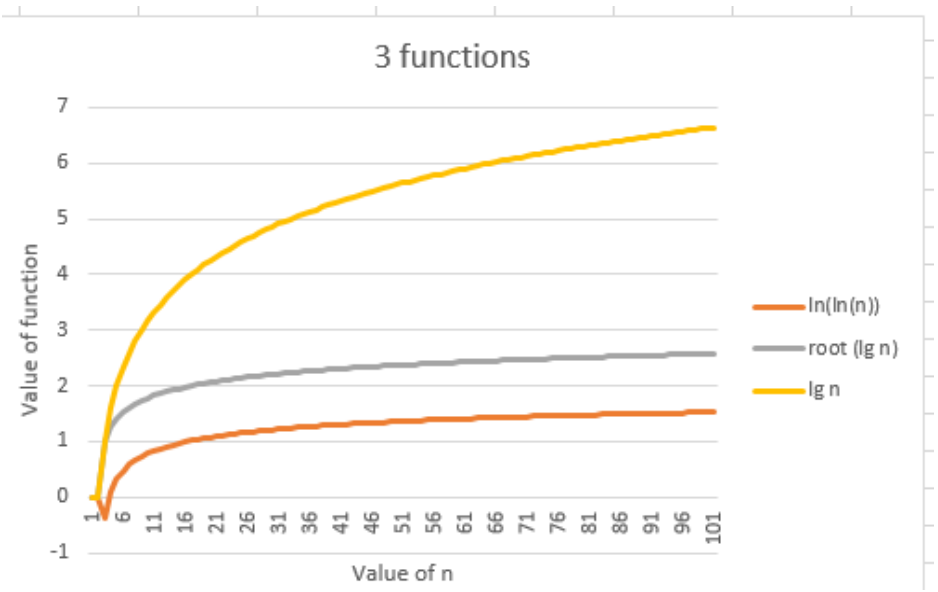
This graph is similar to the above graph except that the exponential powers of this graph are a little lessers than the above graph powers.



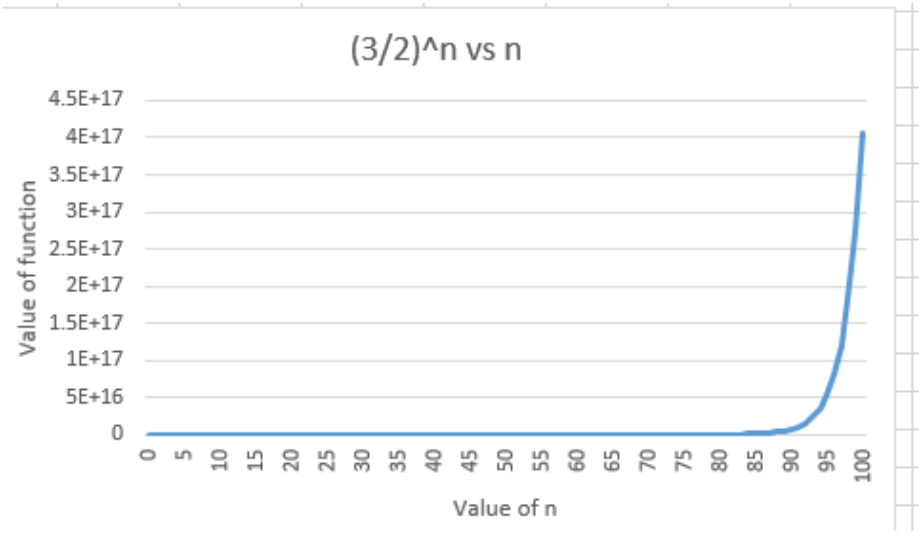
Graph is almost increasing linearly.



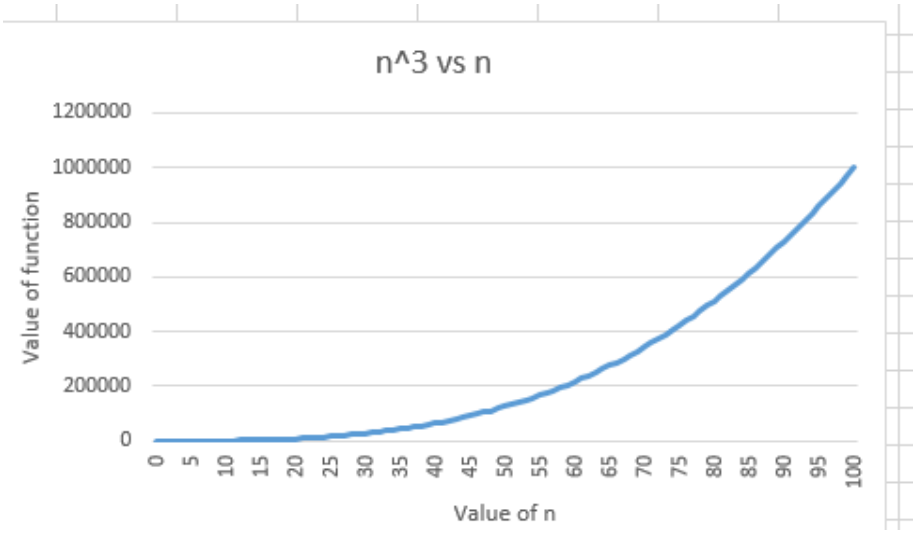
The graph is like a right ward opening parabola with only positive values.



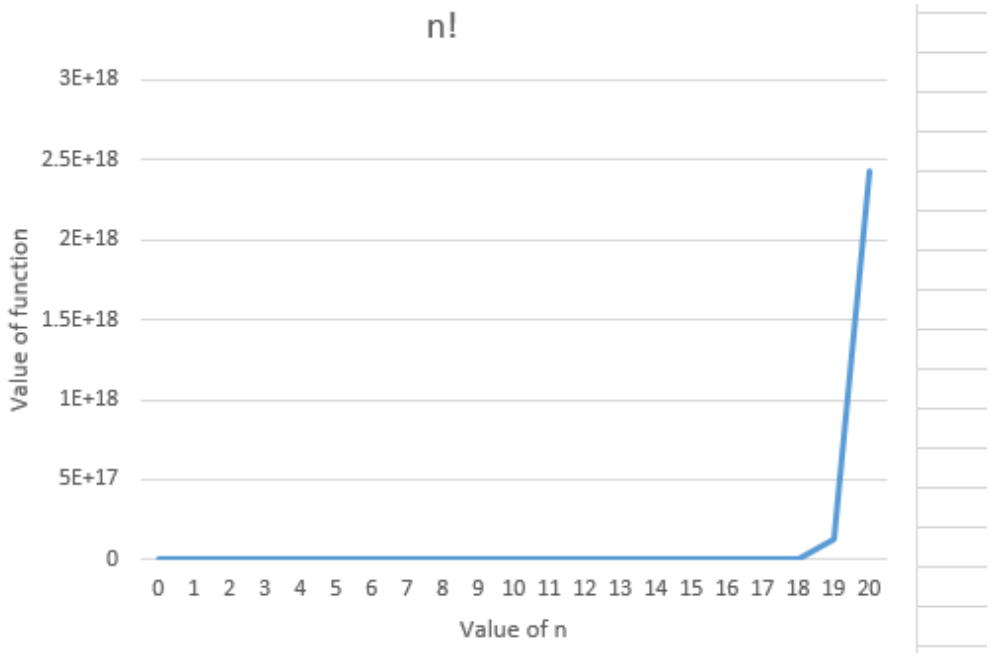
All the three graphs have almost similar curves.(Lg n) has highest values while (ln(ln n)) has least value amongst the three.



The graph is similar to that of n * 2^n graph except that the xponential powers are little lesser here.



The graph is like an upward opening parabola with only positives.



There is rapid increase in value of factorial.

Conclusion:

Program to find the values of the required functions has been implemented in C. After performing this experiment, I learnt how to construct a 2 D plot of the given values in MS Excel and use some basic functions of math.h library of C language.