Name: Kinnari Shah

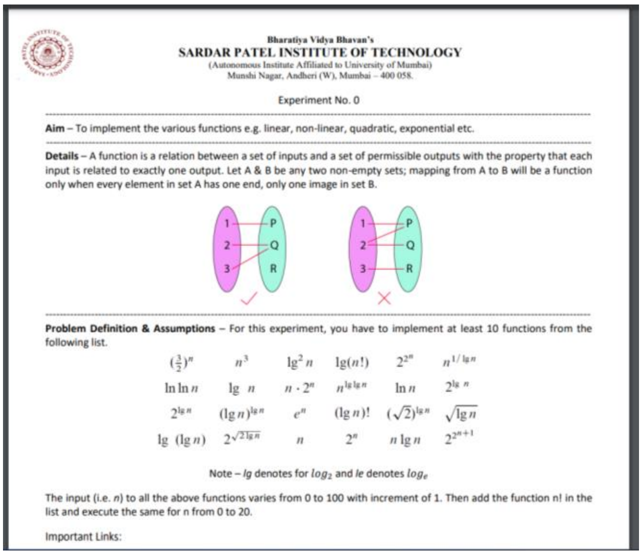
UID: 2021700058

Batch: CSE DS (D4)

Subject: DAA

Experiment No.: 1A

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



Bharatiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.

Experiment No. 0

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

Details – 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.

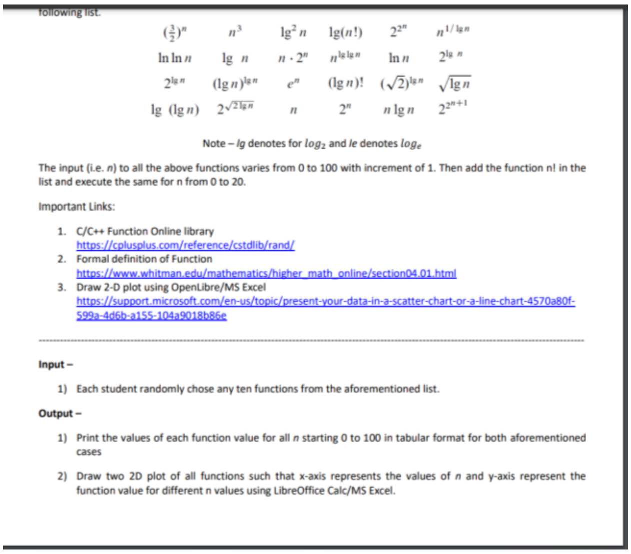
Problem Definition & Assumptions – For this experiment, you have to implement at least 10 functions from the following list.

$(\frac{3}{2})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln \ln n$	$\lg\ n$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	$2^{\lg\ n}$
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$(\lg n)!$	$(\sqrt{2})^{\lg n}$	$\sqrt{\lg n}$
$\lg\ (\lg n)$	$2^{\sqrt{2 \lg n}}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

Note – *lg* denotes for log_2 and *le* denotes log_e

The input (i.e. *n*) to all the above 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.

Important Links:



following list.

$(\frac{3}{2})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln \ln n$	$\lg\ n$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	$2^{\lg\ n}$
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$(\lg n)!$	$(\sqrt{2})^{\lg n}$	$\sqrt{\lg n}$
$\lg\ (\lg n)$	$2^{\sqrt{2 \lg n}}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

Note – *lg* denotes for log_2 and *le* denotes log_e

The input (i.e. *n*) to all the above 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.

Important Links:

1. C/C++ Function Online library
   https://cplusplus.com/reference/cstdlib/rand/
2. Formal definition of Function
   https://www.whitman.edu/mathematics/higher_math_online/section04.01.html
3. Draw 2-D plot using OpenLibre/MS Excel
   https://support.microsoft.com/en-us/topic/present-your-data-in-a-scatter-chart-or-a-line-chart-4570a80f-599a-4d6b-a155-104a9018b86e

Input –

1) Each student randomly chose any ten functions from the aforementioned list.

Output –

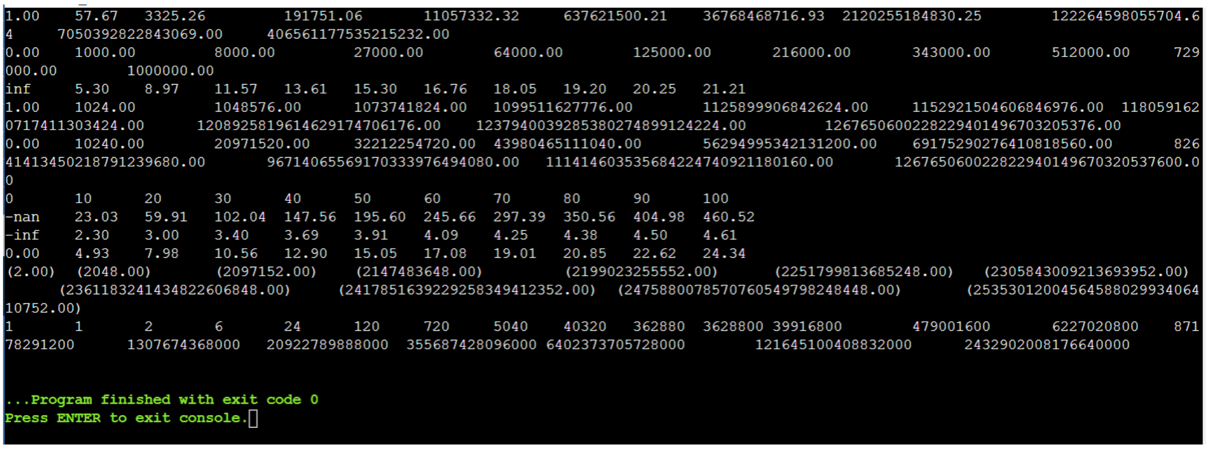
1) Print the values of each function value for all *n* starting 0 to 100 in tabular format for both aforementioned cases

2) Draw two 2D plot of all functions such that x-axis represents the values of *n* and y-axis represent the function value for different n values using LibreOffice Calc/MS Excel.

PROGRAM

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

double f1(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(3.0/2,arr[i]);
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f2(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(arr[i],3);
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f3(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(log(arr[i]),2);
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f4(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(2,arr[i]);
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f5(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(2,arr[i]);
      printf("%.2lf \t",arr[i]*r);
    }
    printf("\n");
}

double f6(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      printf("%d \t",arr[i]);
    }
    printf("\n");
}

double f7(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=arr[i]*log(arr[i]);
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f8(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=log(arr[i]);
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f9(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(2,log(arr[i]));
      printf("%.2lf \t",r);
    }
    printf("\n");
}

double f10(int arr[])
{
    for(int i=0;i<=10;i++)
    {
      double r=pow(2,(arr[i]+1));
      printf("(%.2lf) \t",r);
    }
    printf("\n");
}

long fact(int n)
{
        if(n==0)
        return 1;


        else
        return (n*fact(n-1));
}

int main()
{
    int arr[11]={0,10,20,30,40,50,60,70,80,90,100};
    f1(arr);
    f2(arr);
    f3(arr);

    f4(arr);

    f5(arr);

    f6(arr);

    f7(arr);

    f8(arr);

    f9(arr);

    f10(arr);


    for(int i=0;i<=20;i++)

    {

        fact(i);

        printf("%ld\t",fact(i));

    }

    printf("\n");


    return 0;

}
```

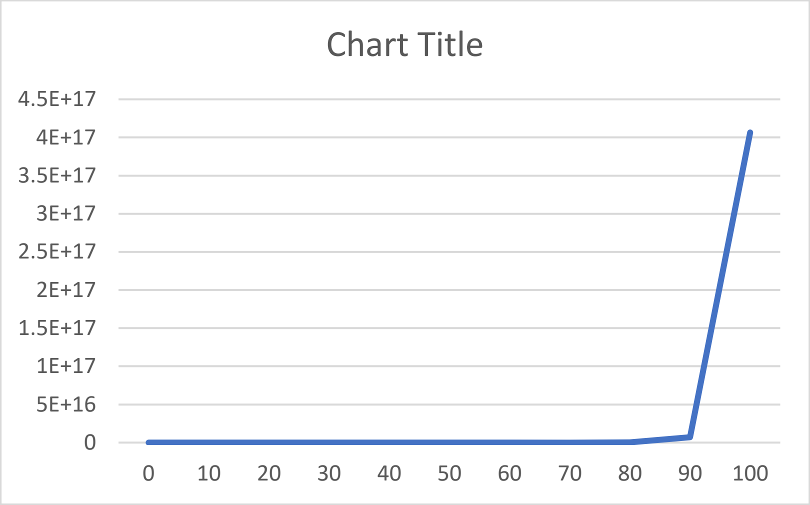
OUTPUT

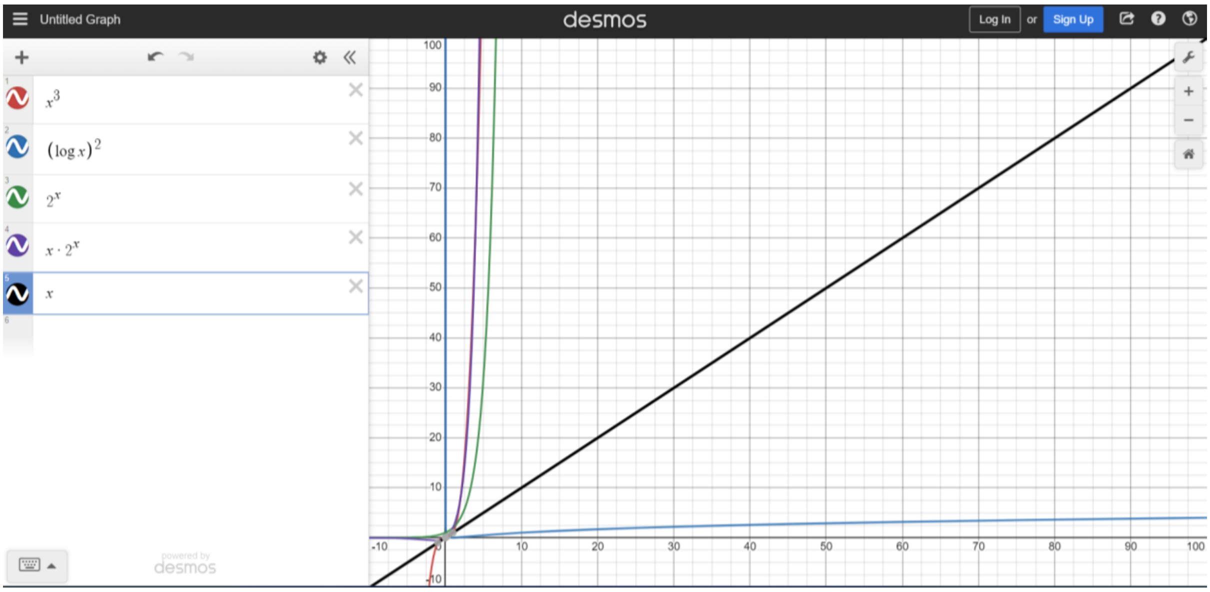


```
1.00    57.67   3325.26         191751.06       11057332.32     637621500.21    36768468716.93  2120255184830.25        122264598055704.6
4      7050392822843069.00     406561177535215232.00
0.00    1000.00         8000.00         27000.00        64000.00        125000.00       216000.00       343000.00       512000.00      729
000.00          1000000.00
inf     5.30    8.97    11.57   13.61   15.30   16.76   18.05   19.20   20.25   21.21
1.00    1024.00         1048576.00      1073741824.00   1099511627776.00        1125899906842624.00     1152921504606846976.00  118059162
0717411303424.00       1208925819614629174706176.00    1237940039285380274899124224.00         1267650600228229401496703205376.00
0.00    10240.00        20971520.00     3221225472O.00  43980465111040.00       5629499534213120O.00    69175290276410818560.00         826
4141345021879123968O.00        96714065569170333976494080.00   1114146035356842247409211801 60.00       12676506002282294014967032053760O.0
0
0       10      20      30      40      50      60      70      80      90      100
-nan    23.03   59.91   102.04  147.56  195.60  245.66  297.39  350.56  404.98  460.52
-inf    2.30    3.00    3.40    3.69    3.91    4.09    4.25    4.38    4.50    4.61
0.00    4.93    7.98    10.56   12.90   15.05   17.08   19.01   20.85   22.62   24.34
(2.00)  (2048.00)       (2097152.00)    (2147483648.00)         (2199023255552.00)      (2251799813685248.00)   (2305843009213693952.00)
       (2361183241434822606848.00)     (2417851639229258349412352.00)  (2475880078570760549798248448.00)       (25353012004564588029934064
10752.00)
1       1       2       6       24      120     720     5040    40320   362880  3628800 39916800        479001600       6227020800     871
78291200        1307674368000   20922789888000  355687428096000 6402373705728000        121645100408832000      2432902008176640000


...Program finished with exit code 0
Press ENTER to exit console.
```

Graphs

1st function





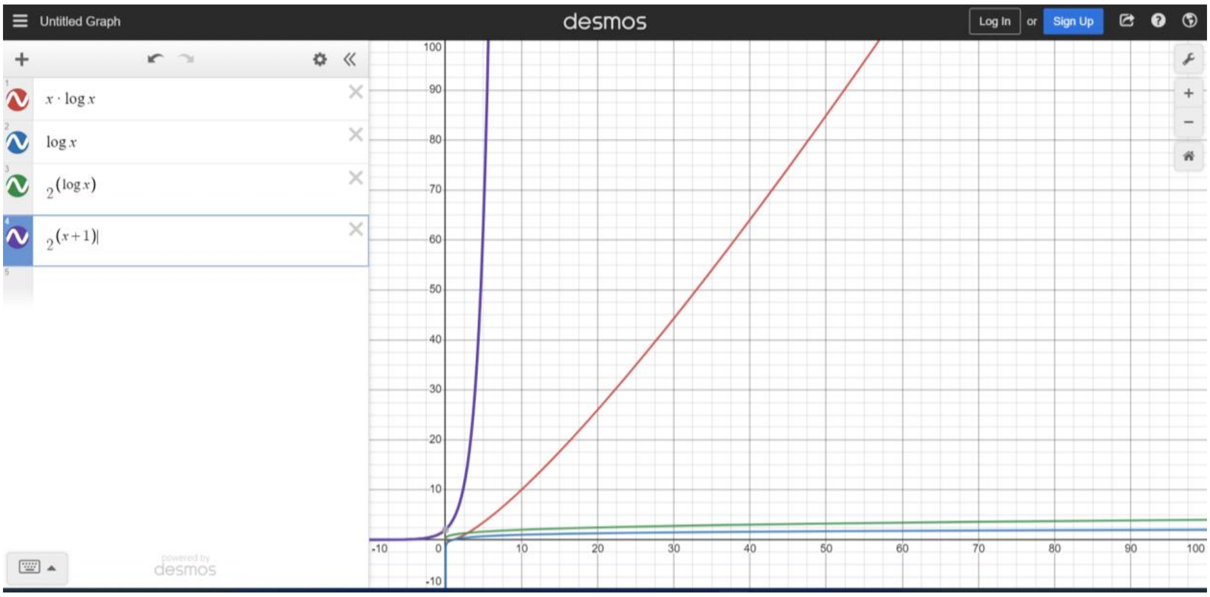
1 x^3

2 $(\log x)^2$

3 2^x

4 $x \cdot 2^x$

5 x



1 $x \cdot \log x$

2 $\log x$

3 $2^{(\log x)}$

4 $2^{(x+1)}$

Conclusion:

In this experiment, we implemented 10 functions in C program and drew a graph for those functions in the values between 0 to 100. From these graphs we observe that some functions are increasing faster than others. For eg 2^(x+1) is increasing faster and is more steeper than x*logx. Also 2^(x+1) is steeper than 2^logx and logx. Also x*2^x is steeper than x, 2^x, x^3 and (logx)^2. All the graphs are increasing in nature from values 0 to 100.