NAME:	MIHIKA CHACHAD
UID:	2021300017
SUBJECT	Design and Analysis of Algorithms
EXPERIMENT NO :	1A
DATE OF PERFORMANCE	24.01.2023
DATE OF SUBMISSION	02.02.2023
AIM:	To implement the various functions e.g., linear, non-linear, quadratic, exponential etc. 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.
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. 1) n 2) n^3 3) log n 4)$n.2^n$ 5)log (log n)

	6) 2^n 7) e^n 8) 3/2.n 9) $(\log n)^{1/2}$ 10) n.(log n) 11) n!
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 = log(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 = n*pow(2,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 = log(log(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 = pow(2,n) 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 = exp(n) (e^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 = 3/2*n iii. Apply a for loop for values of n from 0-100 and print all the values for result. Function 9: i. Initialize variables n and result. ii. result = pow(log(n),0.5) iii. 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 = n*log(n) iii. Apply a for loop for values of n from 0-100 and print all the values for result. Function 11: 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)
```

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

PROGRAM:

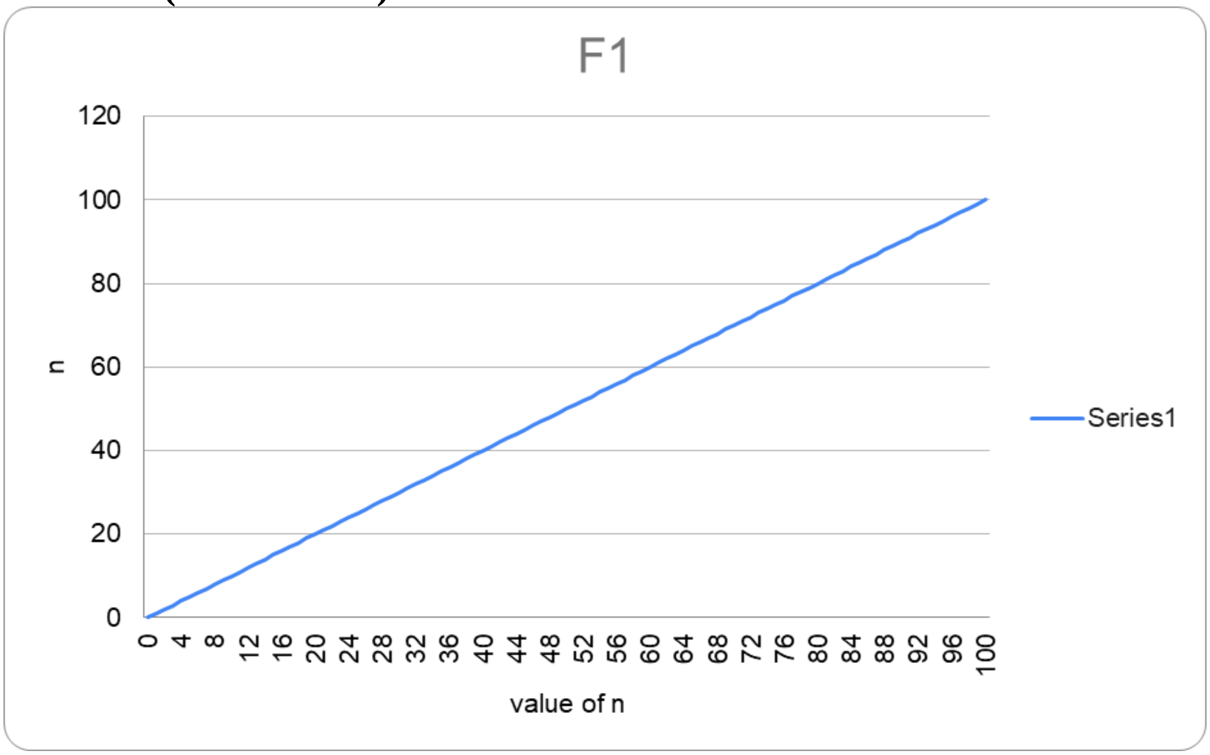
```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    int n,f1,f2,x=0;
    long double f4,f7,f11;
    double f3,f5,f6,f8,f9,f10;
    while(x<=11){
        printf("Enter the function no.:\n");
        scanf("%d",&x);
        if(x==1){
            printf("n\n");
            for(int i=0; i<=100; i++){
                f1=i;
                printf("%d\n",f1);
            }
        }
        if(x==2){
            printf("n^3\n");
            for(int i=0; i<=100; i++){
                f2=i*i*i;
                printf("%d\n",f2);
            }
        }
```

	<pre>if(x==3){
	    printf("log(n)\n");
	    for(int i=0; i<=100; i++){
	        f3=log(i);
	        printf("%.2lf\n",f3);
	    }
	}
	if(x==4){
	    printf("n.2^n\n");
	    for(int i=0; i<=100; i++){
	        f4=i*(pow(2,i));
	        printf("%.Lf\n",f4);
	    }
	}
	if(x==5){
	    printf("(3/2)n\n");
	    for(int i=0; i<=100; i++){
	        f5=1.5*i;
	        printf("%.2lf\n",f5);
	    }
	}
	if(x==6){
	    printf("e^n\n");
	    for(int i=0; i<=100; i++){
	        f6=exp(i);
	        printf("%.2lf\n",f6);
	    }
	}
	if(x==7){
	    printf("2^n\n");
	    for(int i=0; i<=100; i++){
	        f7=(pow(2,i));
	        printf("%.Lf\n",f7);
	    }</pre>

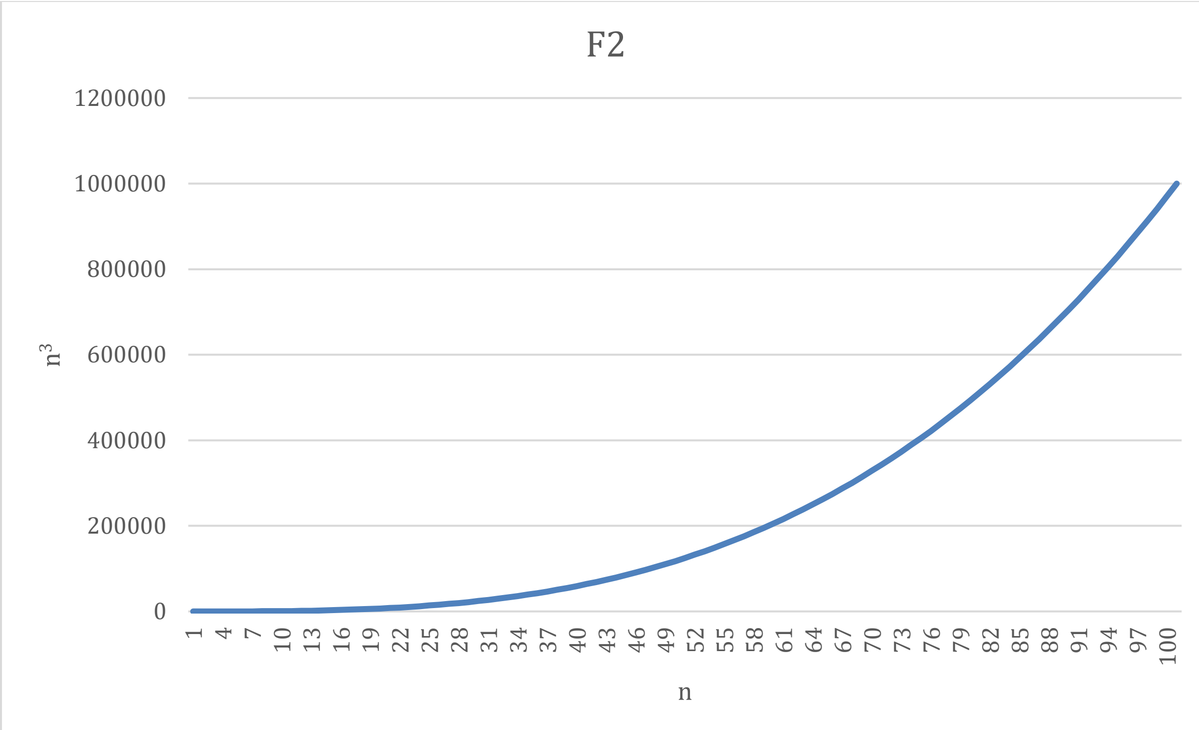
```
        }
        if(x==8){
            printf("log(log(n))\n");
            for(int i=0; i<=100; i++){
                f8=log(log(i));
                printf("%.2lf\n",f8);
            }
        }
        if(x==9){
            printf("(logn)^1/2\n");
            for(int i=0; i<=100; i++){
                f9=pow(log(i),0.5);
                printf("%.2lf\n",f9);
            }
        }
        if(x==10){
            printf("n*log(n)\n");
            for(int i=0; i<=100; i++){
                f10=i*log(i);
                printf("%.2lf\n",f10);
            }
        }
        if(x==11){
            printf("n!\n");
            for(int i=0; i<20; i++){
                f11=factorial(i);
                printf("%.2Lf\n",f11);
            }
        }
        else{
            exit(0);
        }
    }
    return 0;
```

	}

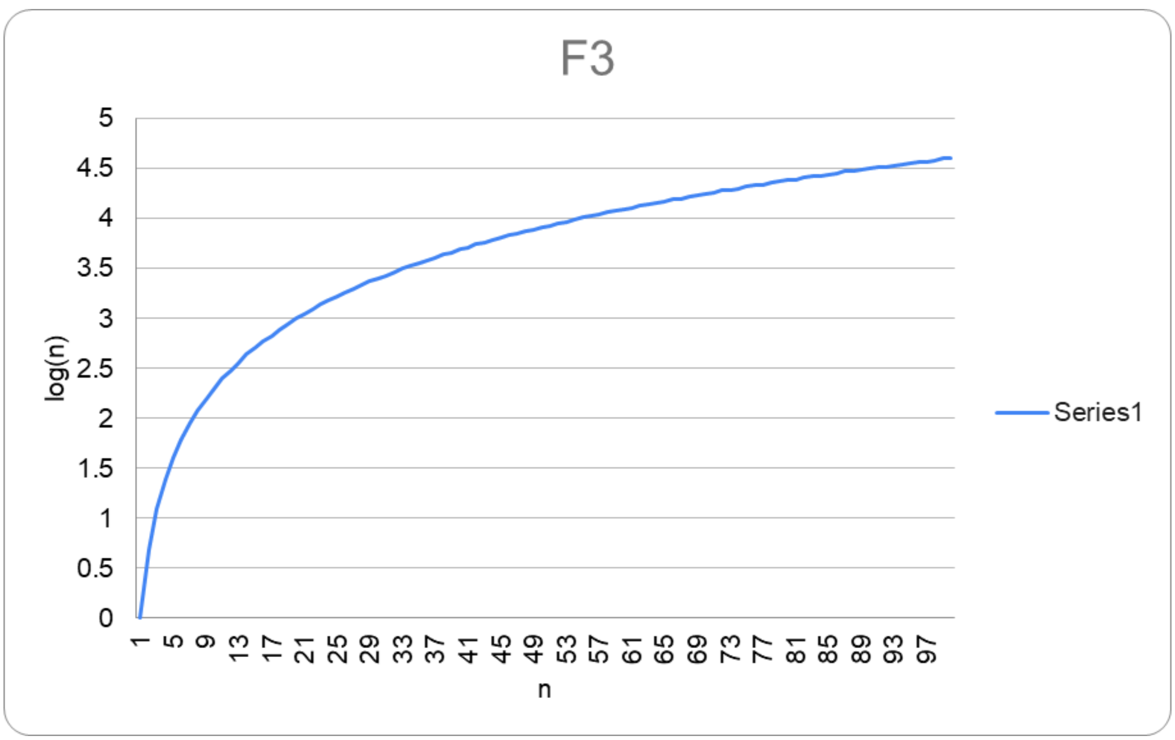
RESULT (SNAPSHOT)



From this graph we observe that, it is a symmetric graph as it is n vs n.

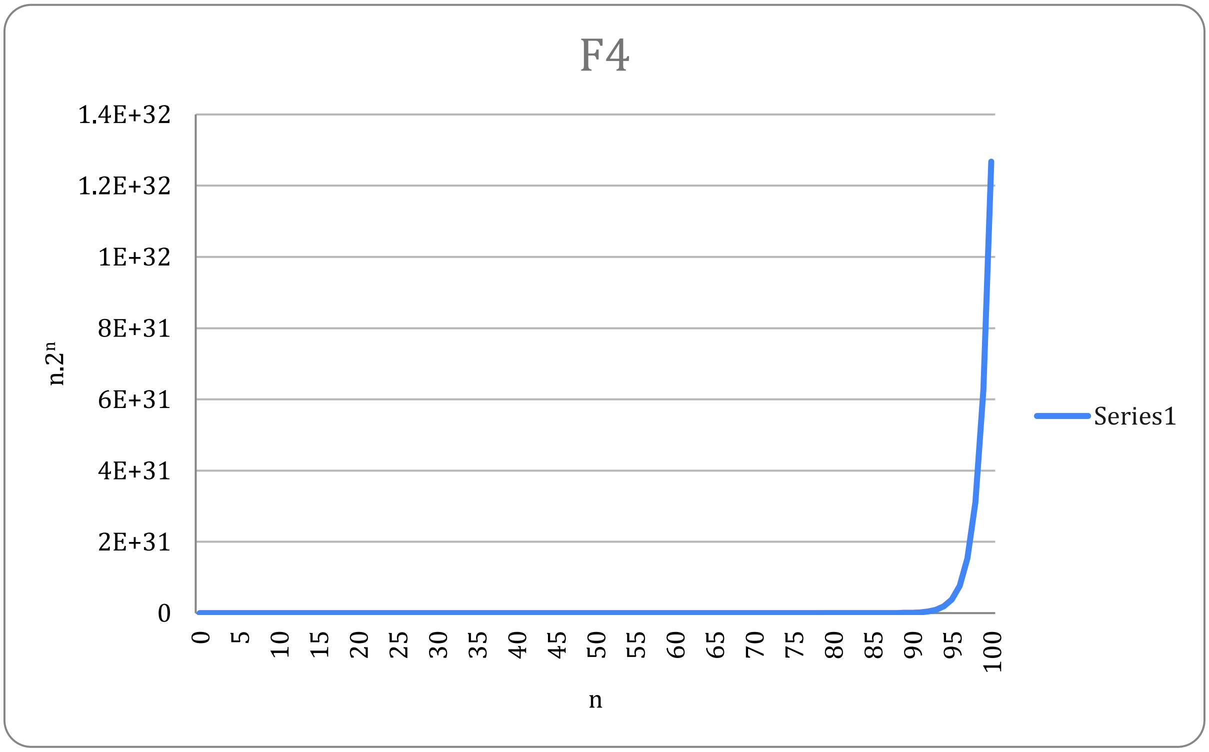


From this graph we observe that, as the value of n increases the graph grows exponentially.

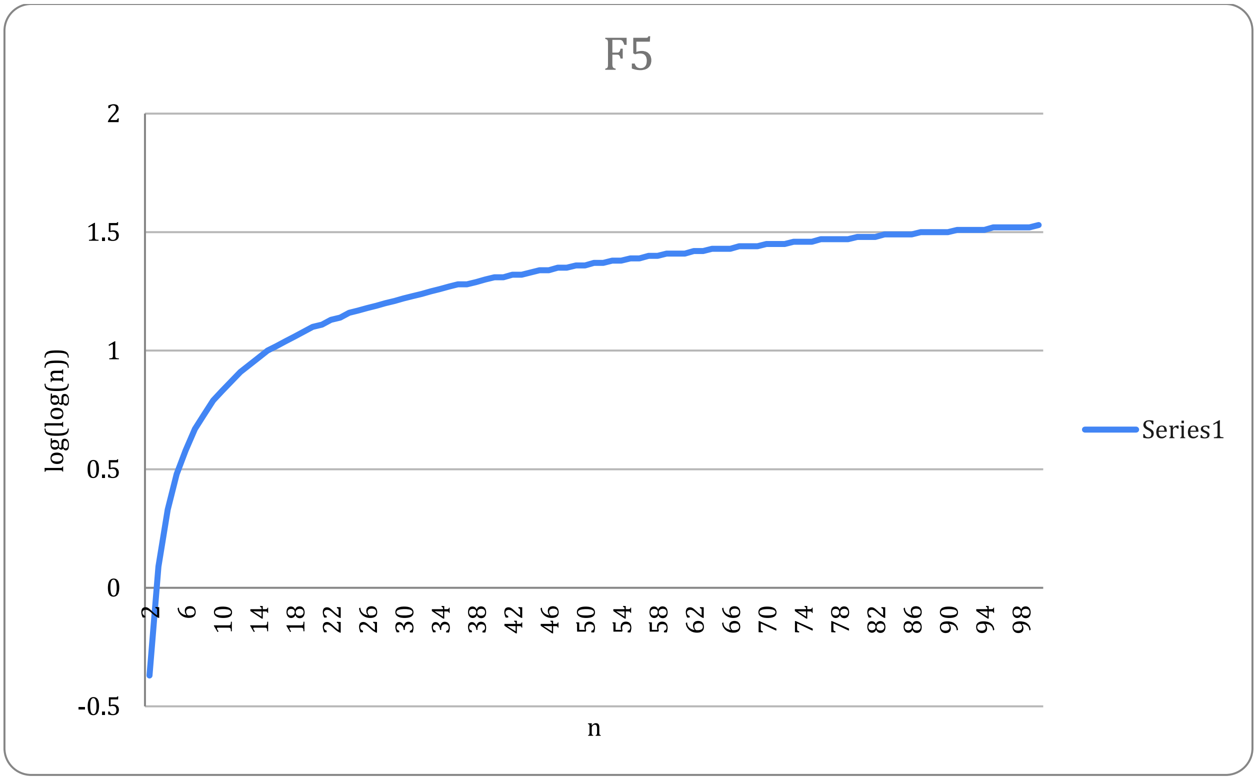


From this graph we observe that, as the value of n increases, the curve grows rapidly then at the later stages, it increases steadily. Here, we don’t take n=0 as log 0 is

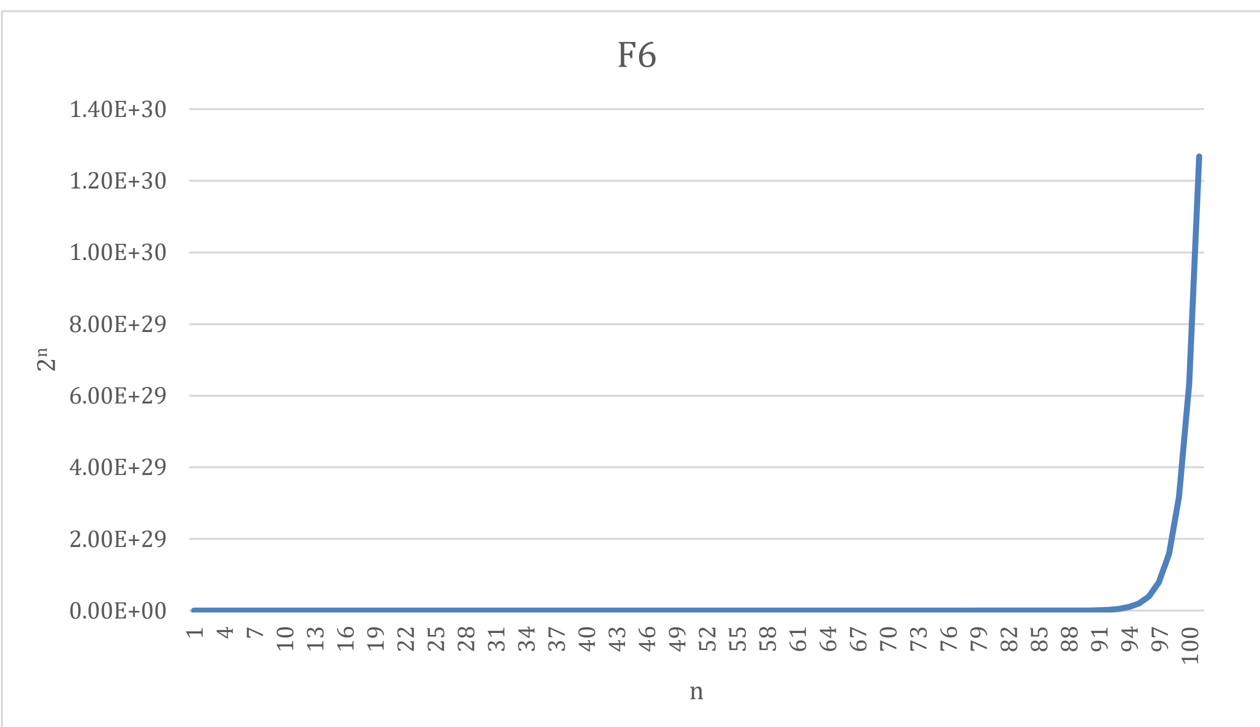
-infinity and cannot be plotted.



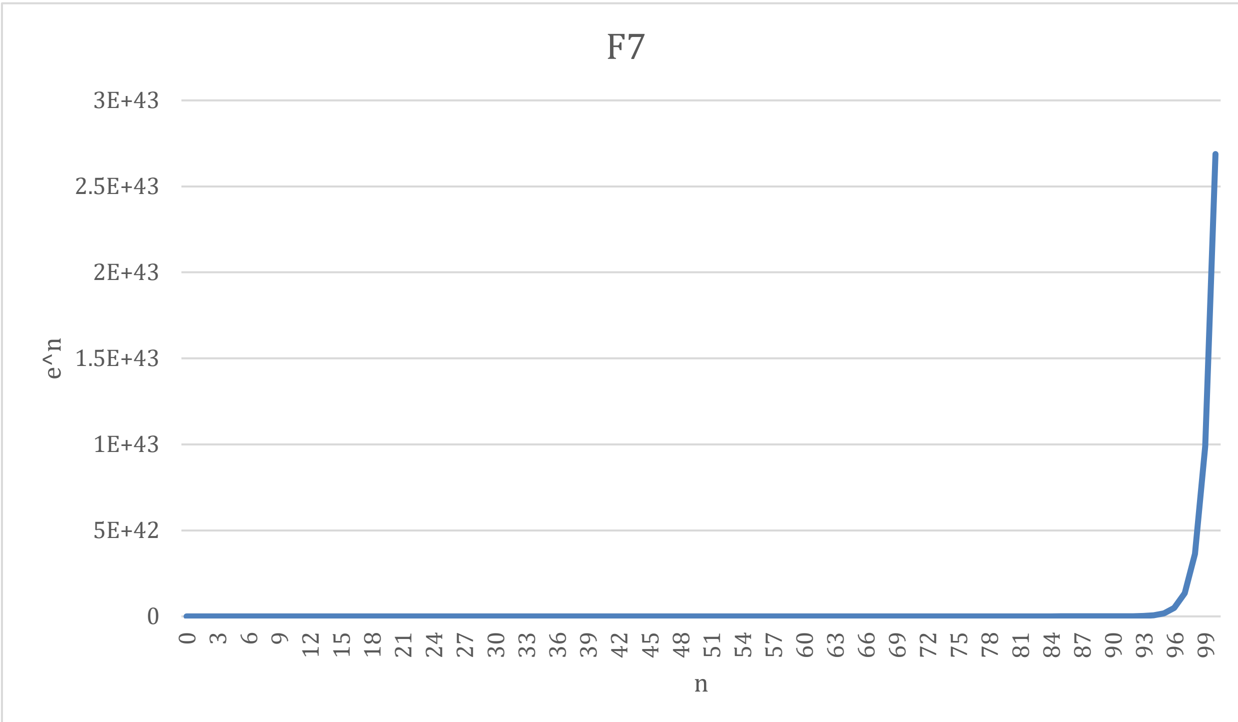
From this graph we observe that, for the values 0-90 the curve increases steadily and sort of linearly but from 90-100 it shoots rapidly.



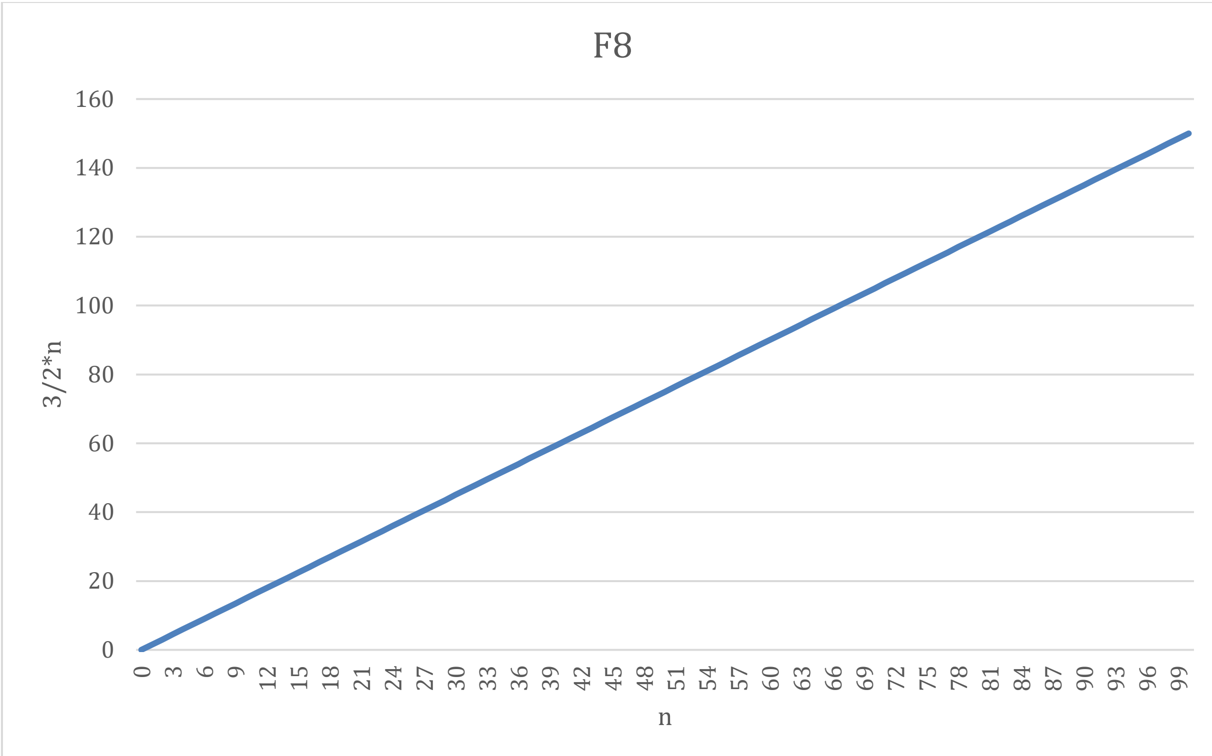
In this graph, we observe that for values 2-25, the graph grows exponentially and then increases linearly and steadily. Here we don't consider n=0,1 as log 0 is not defined and as log 1 is 0 then log(log 1) cannot be defined too.



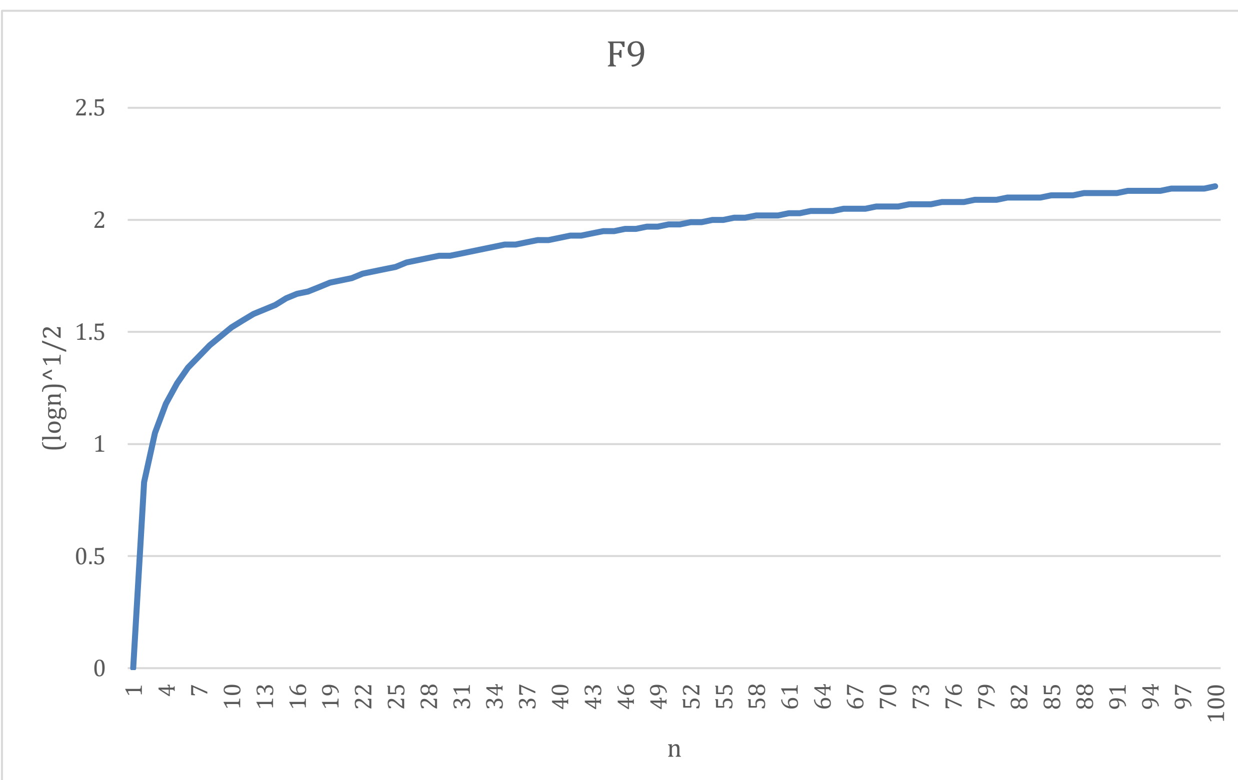
This graph is similar to graph of function no. 4. It increases steadily in the beginning and from 94 or 95 increases sharply.



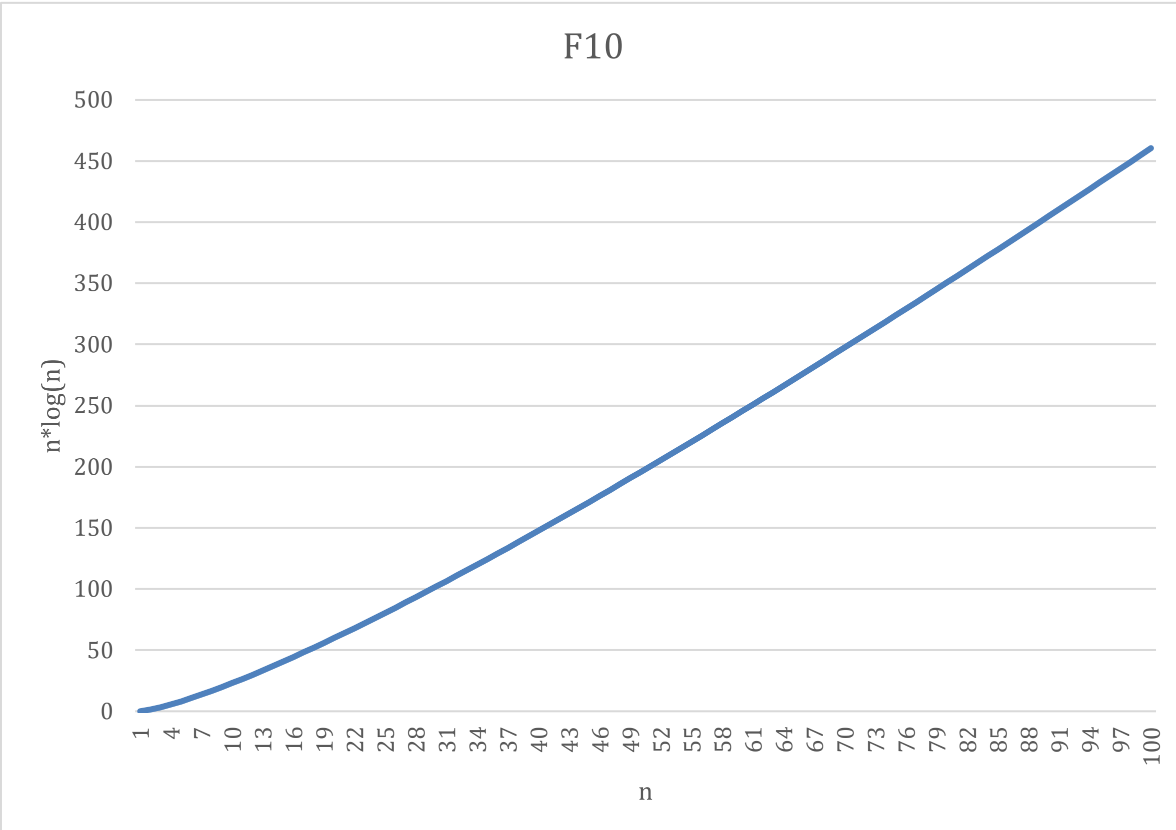
As the previous this graph also increases sharply after 94 or 95.



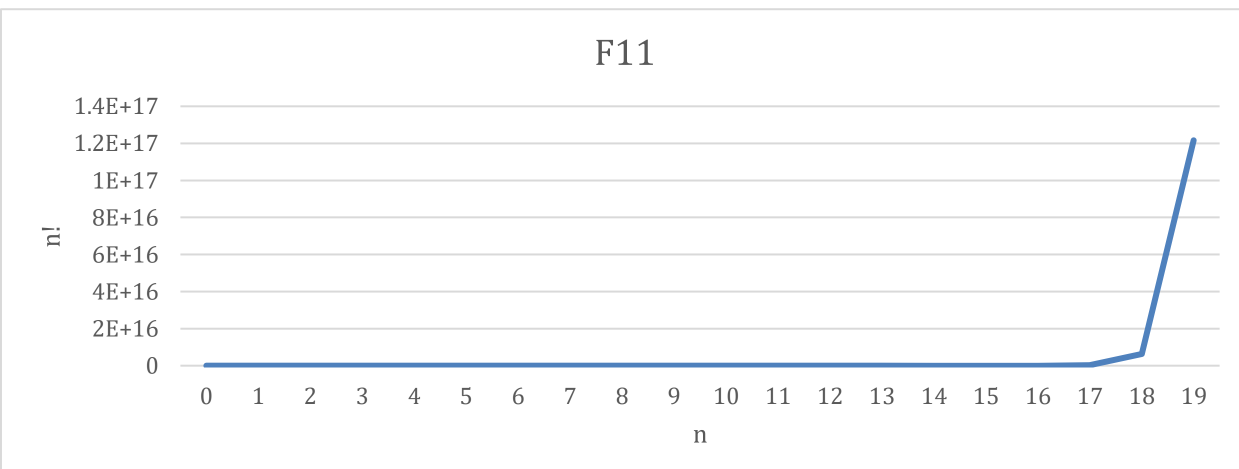
From this graph we can observe that the value of function increases linearly as the value of n increases.



This graph increases rapidly from 1 to 12-13 and then becomes steady.



The curve in this graph increases somewhat linearly as the value of n increases.



The curve for factorial is linear from 0-17 and then shoots rapidly from 18-19.

CONCLUSION:	In this experiment we have plotted graphs for different functions and observed their growth for various cases.