NAME:	Ayush Bodade
UID:	2021300015
SUBJECT	DAA
EXPERIMENT NO :	1
DATE OF PERFORMANCE	23/01/23
DATE OF SUBMISSION	02/02/23
AIM:	To write a program for 10 functions and plot their graph
ALGORITHM:	1.Start 2.Make a menu driven program for all functions 3.In each function definition write the calculation for the specific function in a loop for 0-100 number 4.Ask the user which function he/she wants to print 5.Print the function 6.End

PROGRAM:	#include <stdio.h>
	#include<math.h>
	
	int n;
	void f1()
	{
	int ans[100];
	for(n=0;n<=100;n++)
	{
	ans[n]=n;

```
        printf("%d %d",n,ans[n]);
        printf("\n");
    }
}

void f2()
{
    int ans[100];
    for(n=0;n<=100;n++)
    {
        ans[n]=n*n*n;
        printf("%d %d",n,ans[n]);
        printf("\n");
    }
}
void f3()
{
    double ans[100];
    for(n=0;n<=100;n++)
    {
        ans[n]=pow(2,n);
        printf("%d %.2f",n,ans[n]);
        printf("\n");
    }
}
void f4()
{
    double ans[100];
    for(n=0;n<=100;n++)
    {
        ans[n]=pow(1.5,n);
        printf("%d %.2f",n,ans[n]);
        printf("\n");
    }
```

```
}
double f5()
{
   double ans[100];
   for(n=0;n<=100;n++)
    {
       if (n==0)
        {
           continue;
        }
       ans[n]=log(n);
       printf("%d %.2f",n,ans[n]);
       printf("\n");
    }
}
double f6()
{
   double ans[100];
   for(n=0;n<=100;n++)
    {
       ans[n]=log10(n);
       printf("%d %.2f",n,ans[n]);
       printf("\n");
    }
}
double f7()
{
   double ans[100];
   for(n=0;n<=100;n++)
    {
       ans[n]=sqrt(log2(n));
       printf("%d %.2f",n,ans[n]);
       printf("\n");
    }
```

```
}
double f8()
{
   double ans[100];
   for(n=0;n<=100;n++)
    {
       ans[n]=n*pow(2,n);
       printf("%d %.2f",n,ans[n]);
       printf("\n");
    }
}
double f9()
{
   double ans[100];
   for(n=0;n<=100;n++)
    {
       ans[n]=exp(n);
       printf("%d %.2f",n,ans[n]);
       printf("\n");
    }
}
double f10()
{
   double ans[100];
   for(n=0;n<=100;n++)
    {
       ans[n]=pow(2,log2(n));
       printf("%d %.2f",n,ans[n]);
       printf("\n");
    }
}
double fac(int i)
{
   if (i>=1)
```

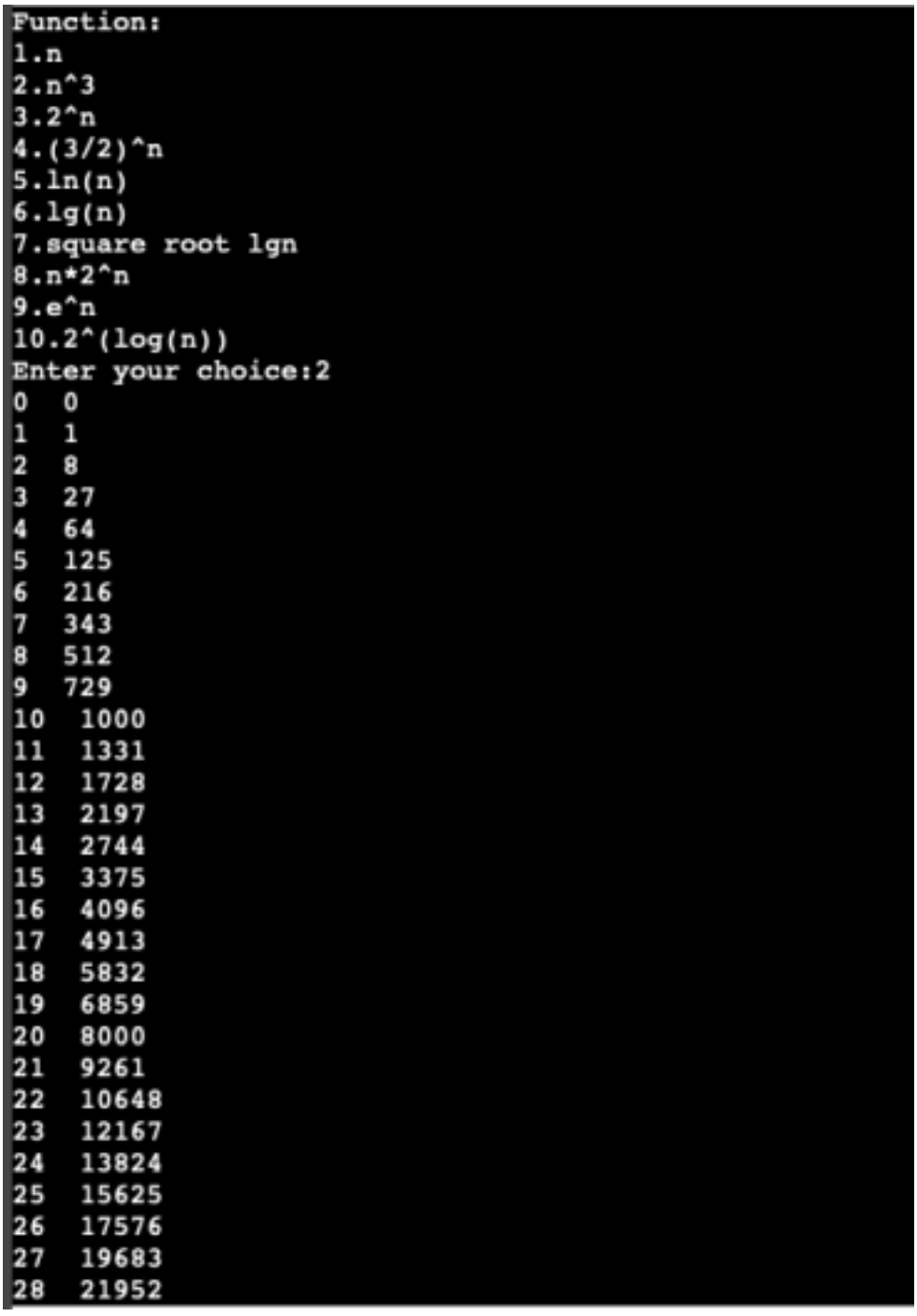
```
        return i*fac(i-1);
    else
        return 1;
}
double f11()
{
    double ans[100];
    for(n=0;n<=100;n++)
    {
        ans[n]=fac(n);
        printf("%.2f",ans[n]);
        printf("\n");
    }
}


int main()
{

printf("Function:\n1.n\n2.n^3\n3.2^n\n4.(3/2)^n\n5.ln(n)\n6.lg(
n)\ n7.square root lgn\n8.n*2^n\n9.e^n\n10.2^(log(n))\n11.n!");
printf("\nEnter your choice:");
    int ch;
    scanf("%d", &ch);
    if(ch==1)
    {
        f1();
    }
    else if(ch==2)
    {
        f2();
    }
    else if(ch==3)
    {
```

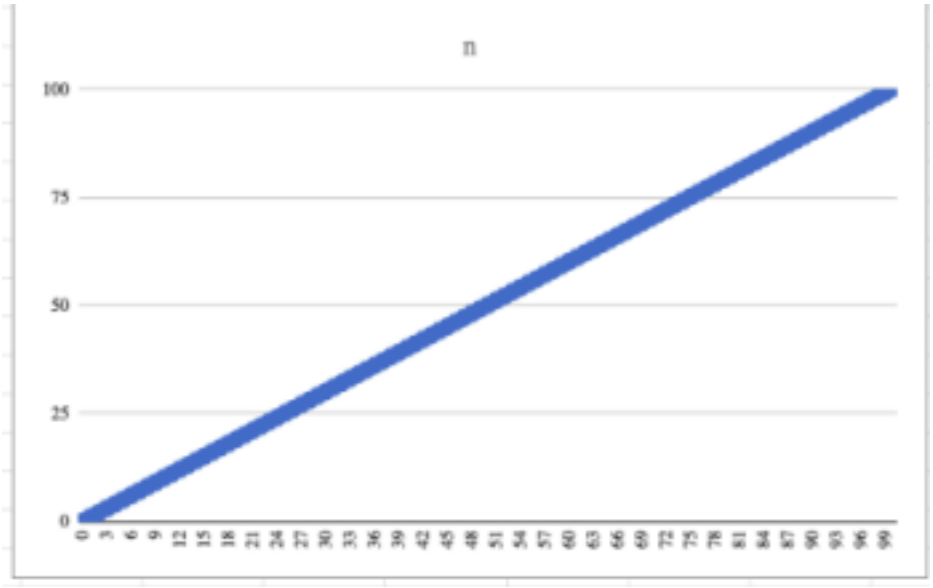
```
        f3();
    }
    else if(ch==4)
    {
        f4();
    }
    else if(ch==5)
    {
        f5();
    }
    else if(ch==6)
    {
        f6();
    }
    else if(ch==7)
    {
        f7();
    }
    else if(ch==8)
    {
        f8();
    }
    else if(ch==9)
    {
        f9();
    }
    else if(ch==10)
    {
        f10();
    }
    else if(ch==11)
    {
        f11();
    }
}
```

RESULT (SNAPSHOT):

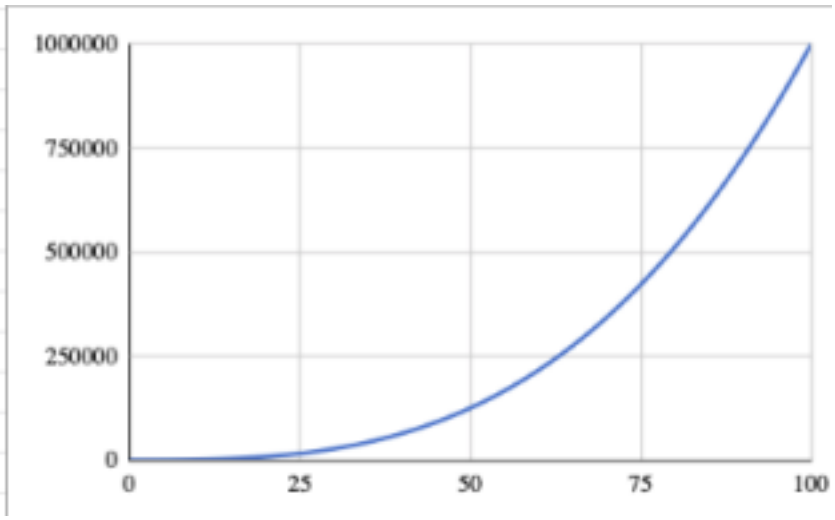


```
Function:
1.n
2.n^3
3.2^n
4.(3/2)^n
5.ln(n)
6.lg(n)
7.square root lgn
8.n*2^n
9.e^n
10.2^(log(n))
Enter your choice:5
1 0.00
2 0.69
3 1.10
4 1.39
5 1.61
6 1.79
7 1.95
8 2.08
9 2.20
10 2.30
11 2.40
12 2.48
13 2.56
14 2.64
15 2.71
16 2.77
17 2.83
18 2.89
19 2.94
20 3.00
21 3.04
22 3.09
23 3.14
24 3.18
25 3.22
26 3.26
27 3.30
```

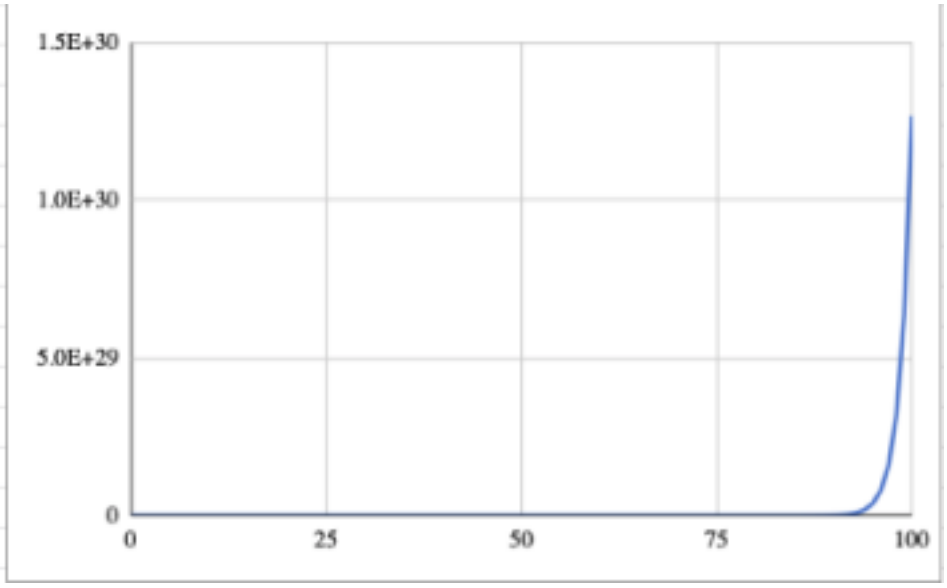
Graphs:



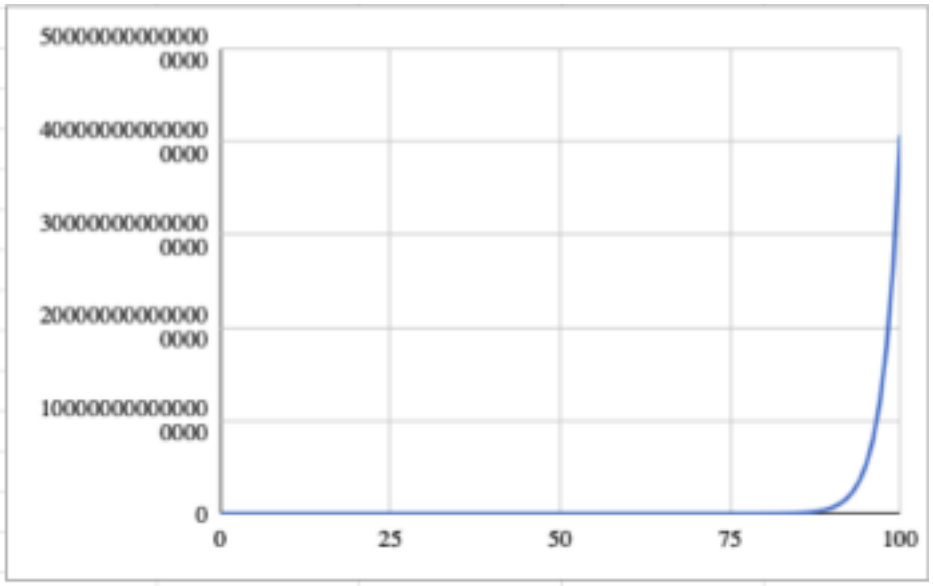
This a n function which gives a linear graph i.e as n increases the value of function increases linearly.



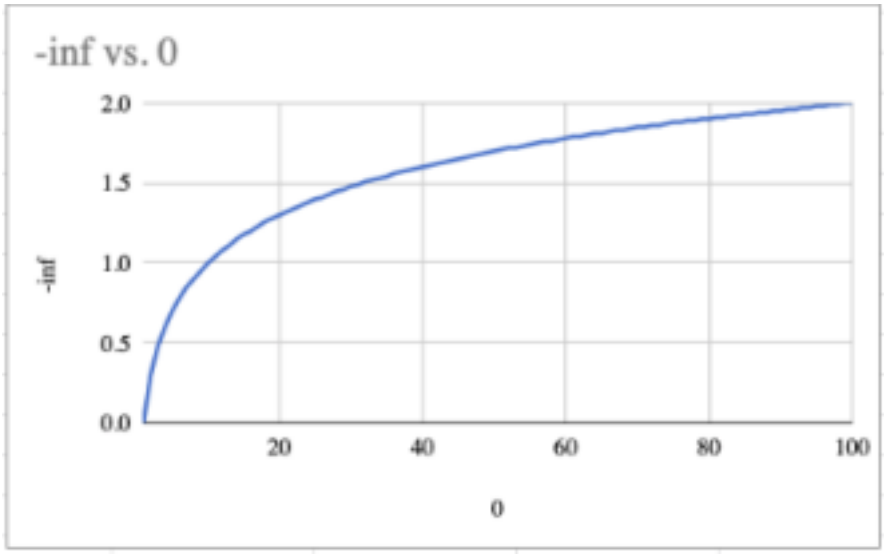
This a 2^n function which gives an exponential graph i.e as n increasing the value of function increases exponentially.



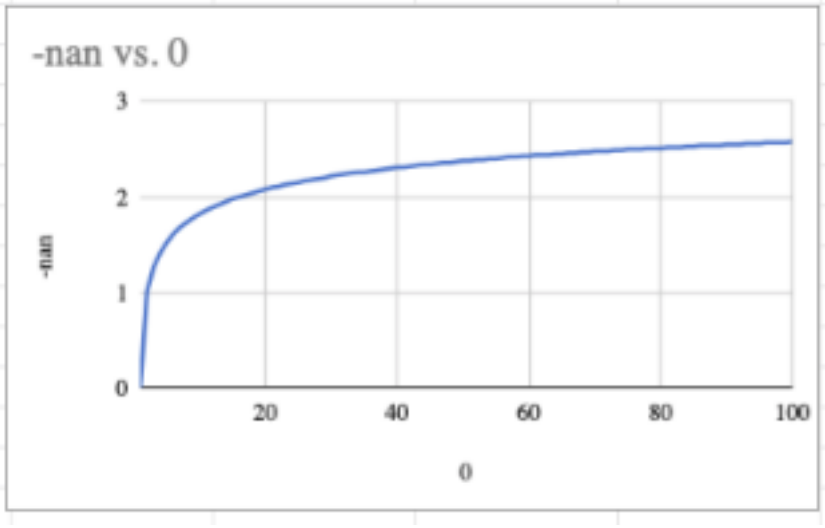
This a (3/2)^n function which gives an exponential graph i.e as n increasing the value of function increases exponentially.



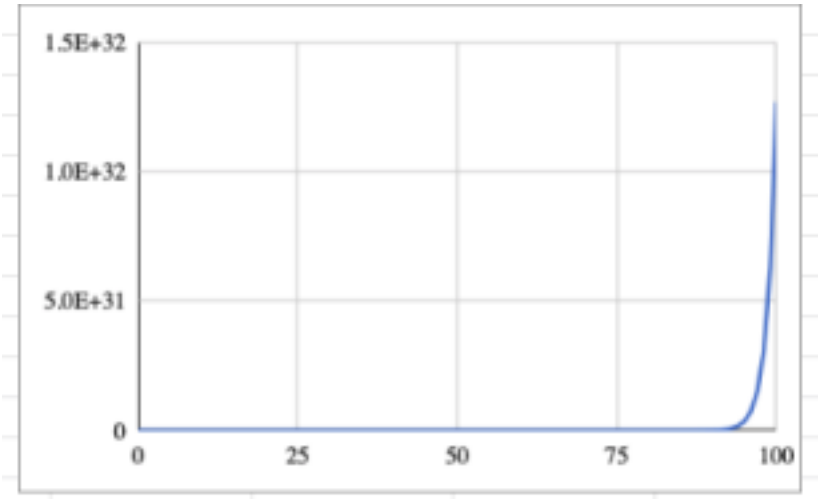
This a ln(n) function which gives an exponential graph i.e as n increasing the value of function increases exponentially.



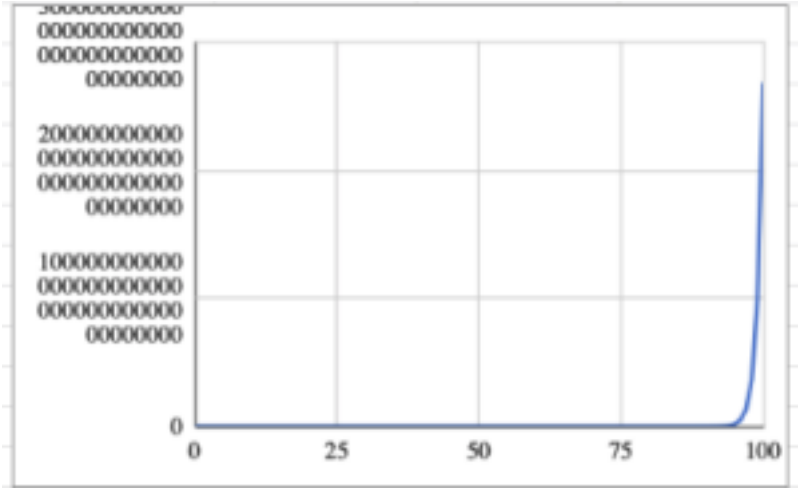
This a lg(n) function which gives a logarithmic graph i.e as n increasing the value of function increases logarithmically.



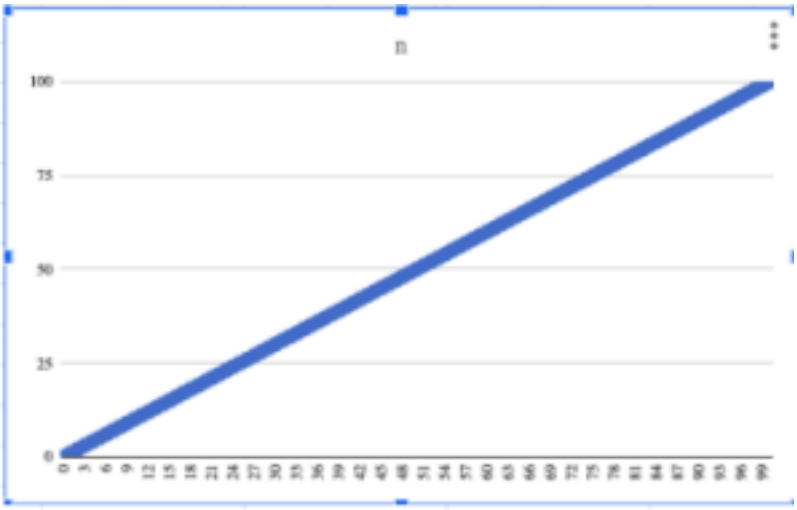
This a square root lgn function which gives a logarithmic graph i.e as n increasing the value of function increases logarithmically.



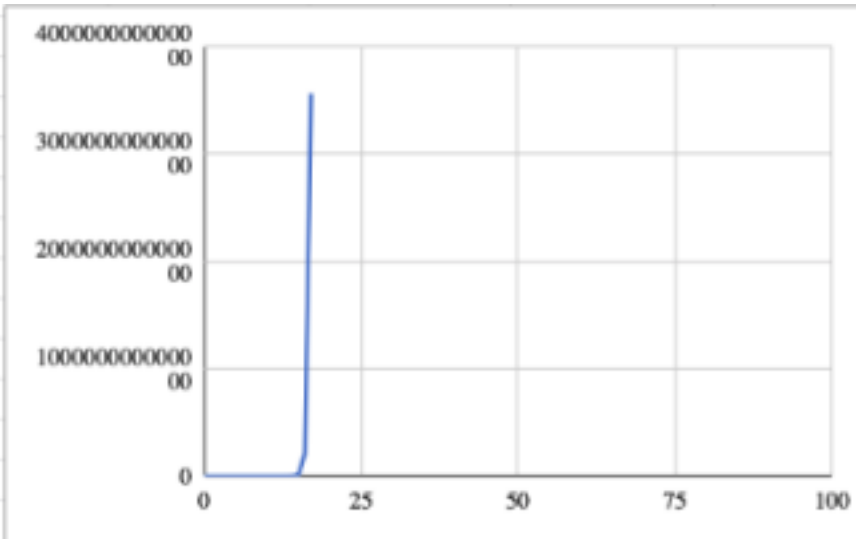
This a n*2^n function which gives a exponential graph i.e as n increasing the value of function increases exponentially.



This an e^n function which gives an exponential graph i.e as n increasing the value of function increases exponentially.



This a 2^(log(n)) function which gives a linear graph i.e as n increases the value of function increases linearly.



This a n! function which gives an exponential graph i.e as n increasing the value of function increases exponentially.

CONCLUSION:	I understood the algorithm behind coding and the use of graph plotting.