*****Computer programme written by TOTON SARKAR*****

```
/* -----------Bisection rule find a root of equation---------- */
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;
float f(float x)
        {
        return(x*x-5*x+6);
        }
int main (void)
{
float a,b,eps,x;
ofstream out("bisec1.out");
cout<<"Enter the accuracy,eps:"<<endl;
cin>>eps;
//**********************
do
{
cout<<"Enter the guess value \n A: ";
cin>>a;
cout<<" B: ";
cin>>b;
}
while(f(a)*f(b)>eps);
cout<<"The excepted guess value A=  "<<a<<" and B= "<<b<<endl;
//*************************
if(f(a)*f(b)>eps)
{
        if(f(a)<eps)
        {
        cout<<"The one of root is : "<<a<<endl;
        }
        if(f(b)<eps)
        {
        cout<<"The one of root is : "<<b<<endl;
        }
}
else
{
        do
        {
        x=(a+b)/2.0;
        if(f(a)*f(x)<eps)
        {
                b=x;
        }
        else if(f(b)*f(x)<eps)
        {
                a=x;
        }
```

```
            out<<x<<"\t"<<f(x)<<endl;
            }
            while(fabs(a-b)>eps);
            cout<<"The one of root is : "<<x<<endl;
}
}
```

Results:

```
toton@toton-VirtualBox ~/sem4/root finding $ g++ bisec1.C -o bisec1.x
toton@toton-VirtualBox ~/sem4/root finding $ ./bisec1.x
Enter the accuracy,eps:
0.0001
Enter the guess value
 A: 1
 B: 3
The excepted guess value A=  1 and B= 3
The one of root is : 1.99225
```

*****Computer programme written by TOTON SARKAR*****

```
/* -----------Newton raphson Methode find a root of equation---------- */
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

float fun(float x)
{
float d;
d=(x*x*x-5*x*x+6);
return d;
}

float fprime(float x)
{
float e;
e=(3*x*x-10*x);
return e;
}
int main(void)
{
float x1,x,fx,fx1,t;
ofstream out1("newrap1.out");
cout<<"enter the desired accuracy:\n";
cin>>t;

cout<<"Enter the Guess value:\n";
cin>>x1;
if(abs(fun(x1))<t)
{
cout<<"The resultent root is : "<<x1<<endl;
}
else
{
cout<<"x\t x1\t |x-x1|"<<endl;
	do
	{
	x=x1;
	fx=fun(x);
	fx1=fprime(x);
	x1=x-(fx/fx1);
	cout<<x<<"\t" <<x1<<"\t"<<abs(x-x1)<<endl;
	out1<<x<<"\t" <<x1<<"\t"<<abs(x-x1)<<endl;
	//out1<<x<<"\t"<<x1<<endl;
	}
	while(abs(fun(x1))>t);

	cout<<"The root is = "<<x1<<"\n";

}
return 0;
}
```

*****Computer programme written by TOTON SARKAR*****

```
/* --Newton raphson Methode find a root of equation  f(x,y)=z^2+1-- */
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

float f(float x,float y)
{
float z=x*x-y*y+1.0;
return(z);
}

float g(float x,float y)
{
float z=2.0*x*y;
return(z);
}

float fx(float x,float y)
{
float z=2.0*x;
return(z);
}

float fy(float x,float y)
{
float z=-2.0*y;
return(z);
}

float gx(float x,float y)
{
float z=2.0*y;
return(z);
}

float gy(float x,float y)
{
float z=2.0*x;
return(z);
}

int main()
{
float x0,x,y,y0,x1,y1,det,h1,h2,x3,y3;
ofstream out1("newrap_z2reg1.out");
ofstream out2("newrap_z2reg2.out");
for(x=-2.0;x<=2.0;x=x+0.01)
{
for(y=-2.0;y<=2.0;y=y+0.01)
{
x3=x;
y3=y;
x0=x;
y0=y;
do
{
det=fx(x0,y0)*gy(x0,y0)-fy(x0,y0)*gx(x0,y0);
h1=(g(x0,y0)*fy(x0,y0)-f(x0,y0)*gy(x0,y0))/det;
h2=(f(x0,y0)*gx(x0,y0)-fx(x0,y0)*g(x0,y0))/det;
x1=x0+h1;
y1=y0+h2;
x0=x1;
y0=y1;
}
while(abs(f(x0,y0))>0.0001 || abs(g(x0,y0))>0.0001);
cout<<"the root is "<<x0<<"  +i("<<y0<<")    "<<det<<endl;

if(abs(x0+0.0)<0.001 && abs(y0+1.0)<0.001)
out1<<x3<<"    "<<y3<<endl;

if(abs(x0+0.0)<0.001 && abs(y0-1.0)<0.001)
out2<<x3<<"  "<<y3<<endl;

}
}
}
```

*****Computer programme written by TOTON SARKAR*****

/* --Newton raphson Methode find a root of equation f(x,y)=z^3+1-- */

```
//root evaluate z^3+1 ------------
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

float f(float x,float y)
        {
        return(x*x*x-3.0*x*y*y+1);
        }
float g(float x,float y)
        {
        return(3.0*x*x*y-y*y*y);
        }
float fx(float x,float y)
        {
        return(3.0*x*x-3.0*y*y);
        }
float fy(float x,float y)
        {
        return(-6.0*x*y);
        }
float gx(float x,float y)
        {
        return(6*x*y);
        }
float gy(float x,float y)
        {
        return(3.0*x*x-3.0*y*y);
        }
int main()
{
float x0,x,y,y0,x1,y1,det,h1,h2,x3,y3;
ofstream out1("newrap_z3reg1.out");
ofstream out2("newrap_z3reg2.out");
ofstream out3("newrap_z3reg3.out");
for(x=-2.0;x<=2.0;x=x+0.01)
{
for(y=-2.0;y<=2.0;y=y+0.01)
{
x3=x;
y3=y;
x0=x;
y0=y;
do
{
det=fx(x0,y0)*gy(x0,y0)-fy(x0,y0)*gx(x0,y0);
h1=(g(x0,y0)*fy(x0,y0)-f(x0,y0)*gy(x0,y0))/det;
h2=(f(x0,y0)*gx(x0,y0)-fx(x0,y0)*g(x0,y0))/det;
x1=x0+h1;
y1=y0+h2;
x0=x1;
y0=y1;
}
while(abs(f(x0,y0))>0.001 || abs(g(x0,y0))>0.001);
cout<<"the root is "<<x0<<"  +i("<<y0<<")    "<<det<<endl;
if(abs(x0-0.50)<0.1 && abs(y0-0.86)<0.1)
out1<<x3<<"  "<<y3<<endl;

if(abs(x0-0.50)<0.1 && abs(y0+0.86)<0.1)
out2<<x3<<"  "<<y3<<endl;

if(abs(x0+1.0)<0.1 && abs(y0-0.0)<0.1)
out3<<x3<<"  "<<y3<<endl;

}
}
return 0;
}
```

*****Computer programme written by TOTON SARKAR*****

/* --Newton raphson Methode find a root of equation f(x,y)=z^4+1-- */

```
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

float f(float x,float y)
{
float z=x*x*x*x + y*y*y*y - 6*x*x*y*y + 1;
return(z);
}

float g(float x,float y)
{
float z=4*(x*x*x*y - x*y*y*y);
return(z);
}

float fx(float x,float y)
{
float z=4*x*x*x - 12*x*y*y;
return(z);
}

float fy(float x,float y)
{
float z=4*y*y*y - 12*y*x*x;
return(z);
}
```

```
float gx(float x,float y)
{
float z=12*x*x*y - 4*y*y*y;
return(z);
}

float gy(float x,float y)
{
float z=4*x*x*x - 12*x*y*y;
return(z);
}

int main()
{
float x0,x,y,y0,x1,y1,det,h1,h2,x3,y3;
ofstream out1("newrap_z4reg1.out");
ofstream out2("newrap_z4reg2.out");
ofstream out3("newrap_z4reg3.out");
ofstream out4("newrap_z4reg4.out");
for(x=-2.0;x<=2.0;x=x+0.01)
{
for(y=-2.0;y<=2.0;y=y+0.01)
{
x3=x;
y3=y;
x0=x;
y0=y;
do
{
det=fx(x0,y0)*gy(x0,y0)-fy(x0,y0)*gx(x0,y0);
h1=(g(x0,y0)*fy(x0,y0)-f(x0,y0)*gy(x0,y0))/det;
h2=(f(x0,y0)*gx(x0,y0)-fx(x0,y0)*g(x0,y0))/det;
x1=x0+h1;
y1=y0+h2;
x0=x1;
y0=y1;
}
while(abs(f(x0,y0))>0.0001 || abs(g(x0,y0))>0.0001);
cout<<"the root is "<<x0<<"  +i("<<y0<<")    "<<det<<endl;

if(abs(x0+0.707)<0.001 && abs(y0+0.707)<0.001)
out1<<x3<<"    "<<y3<<endl;

if(abs(x0+0.707)<0.001 && abs(y0-0.707)<0.001)
out2<<x3<<"  "<<y3<<endl;

if(abs(x0-0.707)<0.001 && abs(y0+0.707)<0.001)
out3<<x3<<"  "<<y3<<endl;


if(abs(x0-0.707)<0.001 && abs(y0-0.707)<0.001)
out4<<x3<<"  "<<y3<<endl;

}
}
}
```

*****Computer programme written by TOTON SARKAR*****

```
/*-----------Newton raphson Methode find a root of equation
f(x)=ax^3+bx^2+cx+d ---------*/

/*
for oscillatory  x^3-ax+b=0; where a^3=2b^2 & x0=0;
and x^3-5x=0; where x=1;
*/
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

float fun(float x,float a,float b,float c,float d)
{
float y;
y=(a*x*x*x+b*x*x+c*x+d);
return y;
}

float fprime(float x,float a,float b,float c)
{
float e;
e=(3*a*x*x+2*b*x+c);
return e;
}
int main(void)
{
float x1,x,fx,fx1,t,a,b,c,d;
ofstream out1("newton_raphson1.out");
cout<<"Enter the value of a,b,c,d"<<endl;
cin>>a>>b>>c>>d;
cout<<"the given eqn is : ("<<a<<")x^3+("<<b<<")x^2+("<<c<<")x+("<<d<<")"<<endl;
cout<<"enter the desired accuracy:\n";
cin>>t;

cout<<"Enter the Guess value:\n";
cin>>x1;
if(abs(fun(x1,a,b,c,d))<t)
{
cout<<"The resultent root is : "<<x1<<endl;
}
else
{
cout<<"THE INTRACT VALUE:"<<endl;
cout<<"x\t x1\t |x-x1|"<<endl;
        do
        {
        x=x1;
        fx=fun(x,a,b,c,d);
        fx1=fprime(x,a,b,c);
        x1=x-(fx/fx1);
        cout<<x<<"\t" <<x1<<"\t"<<abs(x-x1)<<endl;
        out1<<x<<"\t"<<x1<<endl;
        }
        while(abs(fun(x1,a,b,c,d))>t);

        cout<<"The root is = "<<x1<<"\n";

}
return 0;
}
```

*****Computer programme written by TOTON SARKAR*****

```
/*-----------Newton raphson Methode find a root of equation
f(x)=ax^3+bx^2+cx+d ---------*/

/*
for oscillatory  x^3-ax+b=0; where a^3=2b^2 & x0=0;
and x^3-5x=0; where x=1;
*/
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

float fun(float x,float a,float b,float c,float d)
{
float y;
y=(a*x*x*x+b*x*x+c*x+d);
return y;
}

float fprime(float x,float a,float b,float c)
{
float e;
e=(3*a*x*x+2*b*x+c);
return e;
}
int main(void)
{
float x1,x,fx,fx1,t,a,b,c,d;
ofstream out("newrap_gen10.out");
ofstream out1("newrap_gen1.out");
cout<<"Enter the value of a,b,c,d"<<endl;
cin>>a>>b>>c>>d;
cout<<"the given eqn is : ("<<a<<")x^3+("<<b<<")x^2+("<<c<<")x+("<<d<<")"<<endl;
out<<"the given eqn is : ("<<a<<")x^3+("<<b<<")x^2+("<<c<<")x+("<<d<<")"<<endl;
cout<<"enter the desired accuracy:\n";
cin>>t;
out<<"the given desired accuracy:"<<t<<endl;

cout<<"Enter the Guess value:\n";
cin>>x1;
out<<"The given Guess value:"<<x1<<endl;
if(abs(fun(x1,a,b,c,d))<t)
{
cout<<"The resultent root is : "<<x1<<endl;
out<<"The resultent root is : "<<x1<<endl;
}
else
{
cout<<"THE INTRACT VALUE:"<<endl;
cout<<"x\t x1\t |x-x1|"<<endl;
out<<"THE INTRACT VALUE:"<<endl;
out<<"x\t x1\t |x-x1|"<<endl;
        do
        {
        x=x1;
        fx=fun(x,a,b,c,d);
        fx1=fprime(x,a,b,c);
        x1=x-(fx/fx1);
        cout<<x<<"\t" <<x1<<"\t"<<abs(x-x1)<<endl;
        out<<x<<"\t" <<x1<<"\t"<<abs(x-x1)<<endl;
        out1<<x<<"\t"<<x1<<endl;
        }
        while(abs(fun(x1,a,b,c,d))>t);

        cout<<"The root is = "<<x1<<"\n";
        out<<"The root is = "<<x1<<"\n";

}
return 0;
}
```

*****Computer programme written by TOTON SARKAR*****

```
/* -------------Least squrefit of output data set-------------*/
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;

float yfit(float x,float a,float b)
        {
        return(a*x+b);
        }
```

```
int main()
{
int i,n;
float x[100],y[100],sumx=0,sumx2=0,sumy=0,sumxy=0;
float avegx,avegx2,avegy,avegxy,a,b;
ofstream out("leastsqurefit_inout.out");
cout<<"Enter how many data we are fited:"<<endl;
cin>>n;
cout<<"Enter the data Xi & Yi: "<<endl;
for(i=0;i<n;i++)
        {
        cin>>x[i]>>y[i];
        }

for(i=0;i<n;i++)
        {
        sumx=sumx+x[i];
        sumx2=sumx2+x[i]*x[i];
        sumy=sumy+y[i];
        sumxy=sumxy+x[i]*y[i];

        }
//cout<<"sumx = "<<sumx<<"     sumx2 = "<<sumx2<<endl;
//cout<<"sumy = "<<sumy<<"     sumxy = "<<sumxy<<endl;

avegx=sumx/n;
avegx2=sumx2/n;
avegy=sumy/n;
avegxy=sumxy/n;
//cout<<"avegx = "<<avegx<<"     avegx2 = "<<avegx2<<endl;
//cout<<"avegy = "<<avegy<<"     avegxy = "<<avegxy<<endl;
a=(avegxy-avegx*avegy)/(avegx2-avegx*avegx);
b=avegy-a*avegx;


for(i=0;i<n;i++)
        {
        cout<<"X["<<i<<"] = "<<x[i]<<"  Y["<<i<<"] = "<<y[i]<<"\t yfit["<<i<<"]= "<<yfit(x[i], a, b) <<endl;
        out<<x[i] <<"  "<<y[i]<<"  "<<yfit(x[i], a, b) <<endl;
        }
cout<<"The fitting value of a= "<<a<<"   b=  "<<b<<endl;

}
```

*****Computer programme written by TOTON SARKAR*****

/* ------------Least squrefit of in file data set------------*/

```
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;
```

```
//Function define----------------------
float yfit(float x,float a,float b)
        {
        return(a*x+b);
        }
int main(void)
{
int i,n;
float x[100],y[100],sumx=0,sumx2=0,sumy=0,sumxy=0;
float avegx,avegx2,avegy,avegxy,a,b;
ifstream in("leastsqurefit.in");
ofstream out("leastsqurefit_infile.out");
cout<<"Enter how many data we are fited:"<<endl;
cin>>n;
//For input data------------------------
for(i=0;i<n;i++)
        {
        in>>x[i]>>y[i];
        }
//For given data display----------------
cout<<"The given data:"<<endl;
for(i=0;i<n;i++)
        {
        cout<<"X["<<i<<"] = "<<x[i]<<"  Y["<<i<<"] = "<<y[i]<<endl;
        }

for(i=0;i<n;i++)
        {
        sumx=sumx+x[i];
        sumx2=sumx2+x[i]*x[i];
        sumy=sumy+y[i];
        sumxy=sumxy+x[i]*y[i];

        }
avegx=sumx/n;
avegx2=sumx2/n;
avegy=sumy/n;
avegxy=sumxy/n;
a=(avegxy-avegx*avegy)/(avegx2-avegx*avegx);
b=avegy-a*avegx;
cout<<"a= "<<a<<"   b=  "<<b<<endl;
for(i=0;i<n;i++)
        {
cout<<"X["<<i<<"] = "<<x[i]<<"  Y["<<i<<"] = "<<y[i]<<"\t yfit["<<i<<"]= "<<yfit(x[i], a, b)
<<endl;
        out<<x[i] <<"  "<<y[i]<<"  "<<yfit(x[i], a, b) <<endl;
        }
}


/*
input data:
```

```
1	4.8
1.5	6.2
2.0	6.8
2.5	8.2
3.0	8.9
*/
```

*****Computer programme written by TOTON SARKAR*****

```
/*-------------Simpson 1/3 rules Intregrate f(x)=sin(x)----------------*/
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;
float f(float x)
	{
	return(sin(x));
	}
int main(void)
{
int n,i;
float a,b,h,x[100],y[100],s,s1,s2=0,s3=0;
cout<<"Give the \n lower limit: ";
ofstream out1("simp13.out");
cin>>a;
cout<<"Upper limit: ";
cin>>b;
cout<<"Enter the even number of interval:";
cin>>n;
h=(b-a)/n;
for(i=0;i<n;i++)
{
x[i]=a+i*h;
y[i]=f(x[i]);
out1<<x[i]<<" \t"<<y[i]<<endl;
}
s1=f(a)+f(b)+4*f(a+h);
for(i=3;i<=n;i=i+2)
	{
	s2=s2+4*f(a+i*h);
	s3=s3+2*f(a+(i-1)*h);
	}
s=s1+s2+s3;
s=(h/3.0)*s;
cout<<"The resultent integral value: "<<s<<endl;
return 0;
}
```

*****Computer programme written by TOTON SARKAR*****

```
/*-------------Simpson 3/8 rules Intregrate f(x)=sin(x)----------------*/
#include<iostream>
#include<cmath>
```

```
#include<fstream>
using namespace std;
float f(float x)
    {
    return(sin(x));
    }
int main(void)
{
int n,i;
float a,b,h,x[100],s,s1,s2=0,s3=0,y[100];
ofstream out1("simp38.out");
cout<<"Give the \n lower limit: ";
cin>>a;
cout<<"Upper limit: ";
cin>>b;
cout<<"Enter the even number of interval:";
cin>>n;
h=(a+b)/n;
s1=f(a)+f(b);
for(i=0;i<n+1;i++)
{
x[i]=a+i*h;
y[i]=f(x[i]);
out1<<x[i]<<" \t "<<f(x[i])<<endl;
}
for(i=1;i<=(n-1);i++)
{
    //x=a+i*h;
    if(i%3==0)
    {
        s2=s2+2*f(x[i]);
    }
    else
    {
        s3=s3+3*f(x[i]);
    }
}
s=s1+s2+s3;
s=((3.0*h)/8.0)*s;
cout<<"The resultent integral value: "<<s<<endl;
return 0;
}
```

*****Computer programme written by TOTON SARKAR*****

-------------Trapezoidal Methods Intregrate f(x)=sin(x)----------------/

```
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;
float f(float x)
    {
    return(sin(x));
```

```
        }
int main(void)
{
int N,i;
float a,b,sum=0,area,h,x;
cout<<"Give the \n lower limit: ";
cin>>a;
cout<<"Upper limit: ";
cin>>b;
cout<<"Enter the value os N:";
cin>>N;
h=(b-a)/N;
for(i=1;i<N;i++)
        {
        x=a+h*i;
        sum=sum+f(x);
        }
area=h*(f(a)+f(b)+2*sum)/2;
cout<<"The resultent area: "<<area<<endl;
}
```

*****Trapezoidal Rule intregrate 0 to inf of exp(-qx)/sqrt(x) *********

```
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;
//*******************
float f(float z,float q)
        {
        float p= 2.0*exp(-q*z*z);
        return p;
        }
//***********************
float sumt(float xo,float xn,int n,float q)
{
int  i;
float sum=0;
float h=(xn-xo)/n;

for(i=1;i<n;i++)
sum = sum + f(xo+i*h,q);

sum = (h/2.0) * ( f(xo,q)+f(xn,q) +2.0*sum );

return sum;
}
//**********************
int main(void)
{
int n;
float xo,xn,I,I1=0,error,q;
//ofstream out("trap2.out");
cout<<"Enter the lower & upper limit : "<<endl;
cin>>xo>>xn;
cout<<"Enter How many division you want to do? ";
cin>>n;
cout<<"Enter the value of q= ";
cin>>q;
//******
do
{
I = sumt(xo,xn,n,q);
error = abs(I-I1);
I1=I;
xn=xn+1.0;
}while(error>0.00001);
cout<<"q= "<<q<<"\nintregral value theoritical: "<<sqrt(M_PI/q)<<" Expt: "<<I1<<endl;

return 0;
}
```

Results:

```
Enter the lower & upper limit :
0
1
Enter How many division you want to do? 100
Enter the value of q= 20
q= 20
intregral value theoritical: 0.396333 Expt: 0.396333
```

-----------Montecarlo Methods Intregrate f(x)=sin(x) for 0 to pai/2-------------/

```
#include<iostream>
#include<fstream>
#include<cstdlib>
#include<ctime>
#include<cmath>
using namespace std;
float f(float x)
        {
        return(sin(x));
        }
int main()
{
int n,Ncount=0,i;
float xi,yi,p,a,b,ymax=0,x,sum;
ofstream out1("montea.out");
ofstream out2("monteb.out");
cout<<"give the lower & upper limit :"<<endl;
cin>>a>>b;
cout<<"Enter how many number are to be generated: ";
cin>>n;
for(x=a;x<=b;x+=0.001)
{
        if(f(x)>ymax)
        {
        ymax=f(x);
        }
}
cout<<"the value of ymax: "<<ymax<<endl;
time_t t;
srand (time(&t));
for(i=0;i<n;i++)
        {
        p=rand();
        xi=(p/(float(RAND_MAX)))*b;
        p=rand();
        yi=(p/(float(RAND_MAX)))*ymax;
        out1<<xi<<"\t"<<yi<<endl;
        if(yi<=f(xi))
                {
                Ncount++;
                out2<<xi<<"\t"<<yi<<endl;
                }

        }
cout<<"Ncount: "<<Ncount<<endl;
sum=(b*ymax*Ncount)/(n*1.0);
cout<<"the value of Intregration: "<<sum<<endl;
return 0;
}
```

Results output:

```
give the lower & upper limit :
0
3.1
Enter how many number are to be generated: 100000
the value of ymax: 1
Ncount: 64283
the value of Intregration: 1.99277
```

----------Montecarlo Methods Intregrate f(x)=sin(x) for all interval-------------/

```
#include<iostream>
#include<fstream>
#include<ctime>
#include<cstdlib>
#include<cmath>
using namespace std;
float f(float x) //For function define--------
        {
        float d=sin(x);
        return d;
        }
float ymax(float a,float b) //For evaluate y maximum values------
        {
```

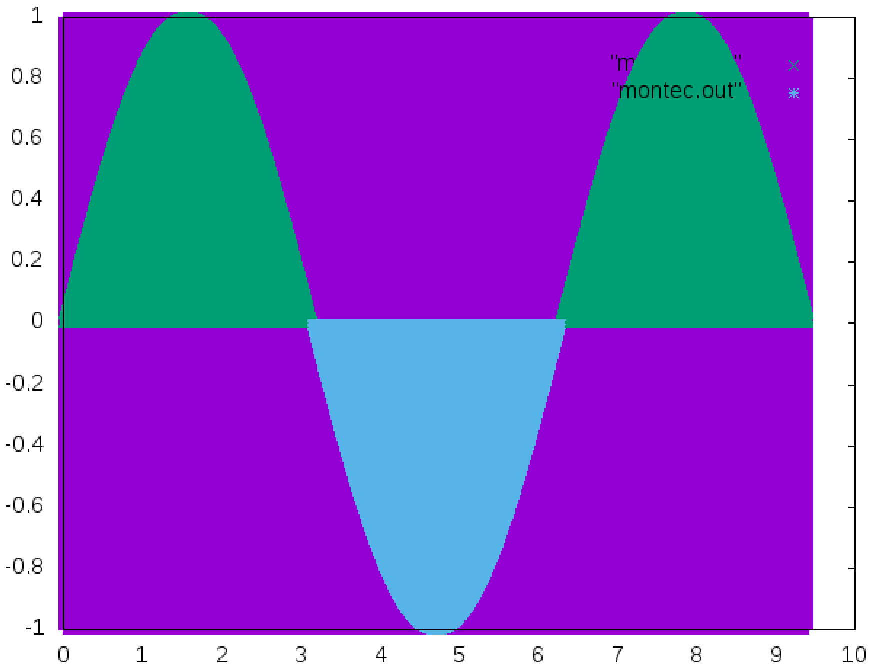
```
        float ymax=0,x;
        for(x=a;x<=b;x+=0.01)
        {
        if(f(x)>ymax)
                {
                ymax=f(x);
                }
        }
        return(ymax);
        }
float ymin(float a,float b) //For evaluate y minimum values------
        {
        float ymin=0,x;
        for(x=a;x<=b;x+=0.01)
        {
        if(f(x)<ymin)
                {
                ymin=f(x);
                }
        }
        return(ymin);
        }
int main() //For main programme-----
{
int n,n1=0,n2=0,i;
float a,b,max,min,xi,yi,x,y,area;
ofstream out("montea.out");
ofstream out1("monteb.out");
ofstream out2("montec.out");
cout<<"Enter the valu of lower & upper limit:"<<endl;
cin>>a>>b;
cout<<"How many point are given:";
cin>>n;
max=ymax(a,b);
min=ymin(a,b);
cout<<"ymax: "<<max<<"  ymin: "<<min<<endl;
time_t t;
srand (time (&t));
for(i=0;i<=n;i++)
{
x=rand()/(float (RAND_MAX));
xi=a+(b-a)*x;
y=rand()/(float (RAND_MAX));
yi=min+(max-min)*y;
out<<xi<<"\t "<<yi<<endl;
if(yi>0 && yi<f(xi))
        {
        n1++;
        out1<<xi<<"\t "<<yi<<endl;
        }
if(yi<0 && yi>f(xi))
        {
```

```
        n2++;
        out2<<xi<<"\t "<<yi<<endl;
        }
}
cout<<"n1= "<<n1<<"  n2=  "<<n2<<endl;
area=((b-a)*(n1-n2)*(max-min))/n;
cout<<"The resultent intreagration: "<<area<<endl;
return 0;
}
```

Results:

```
 toton@toton-VirtualBox ~/trial $ g++ monte1.C -o monte1.x
toton@toton-VirtualBox ~/trial $ ./monte1.x
Enter the valu of lower & upper limit:
0
9.42
How many point are given:10000000
ymax: 1  ymin: -0.999997
n1= 2123032  n2=  1061394
The resultent intreagration: 2.00012

gnuplot> set term png
Terminal type set to 'png'
Options are 'nocrop enhanced size 640,480 font
"/usr/share/fonts/truetype/liberation/LiberationSans-Regular.ttf,12" '
gnuplot> set output "monte.png"
gnuplot> plot "montea.out","monteb.out","montec.out"
```



--Solve 1st order Differential equation by Rungee-Kutta Order four method --/

```
#include<iostream>
#include<cmath>
#include<fstream>

using namespace std;
```

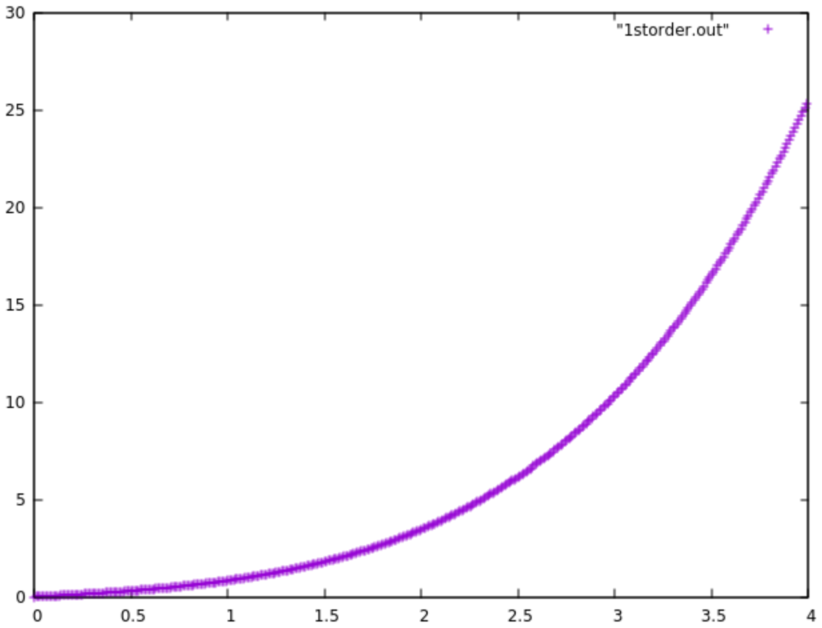
```
float f(float x,float y)
        {
        float p=x*x+1;
        //float p=(y*y-x*x)/(y*y+x*x);
        return (p);
        }

int main()
{
float j,n,x,xn,xf,h,k1,k2,k3,k4,y,yn;
ofstream out("1storder.out");
cout<<"Enter h & xn & yn: "<<endl;
cin>>h>>xn>>yn;
cout<<"for which x I want the value of y???"<<endl;
cin>>x;
n=(x-xn)/h;
for(j=0;j<n;j++)
{
k1=h*f(xn,yn);
k2=h*f((xn+h/2.0),(yn+k1/2.0));
k3=h*((xn+h/2.0),(yn+k2/2.0));
k4=h*((xn+h),(yn+k3));
y=yn+(k1+2*k2+2*k3+k4)/6.0;
out<<xn<<"\t"<<yn<<endl;
xn=xn+h;
yn=y;
}
cout<<"the value of y: "<<xn<<"\t"<<yn<<endl;
return 0;
}
```

Results:

```
toton@toton-VirtualBox ~/trial2/diff_rungekutta $ g++ -o Wall 1storder.C -o 1storder.x
toton@toton-VirtualBox ~/trial2/diff_rungekutta $ ./1storder.x
Enter h & xn & yn:
0.01
0
0
for which x I want the value of y???
4
the value of y: 4       25.5114

gnuplot> plot "1storder.out"
gnuplot>
```



**Solve 2st order Differential equation y^..+siny=0, y^.=dy/dt by Rungee-Kutta Order four cupple variable method **

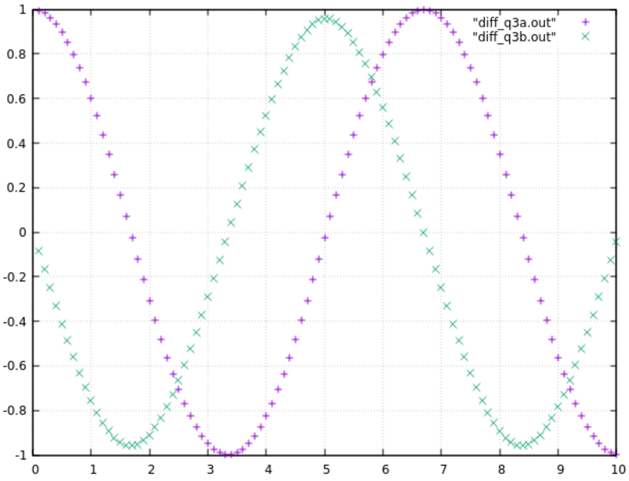
```
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;
//*************************************
float gx(float x,float y,float z)
{
float q = z;
return q;
}
//***********************************
float fx(float x,float y,float z)
{
float q =-sin(y);
return q;
}
//***********************************
int main(void)
{
ofstream out1("diff_q3a.out");
ofstream out2("diff_q3b.out");
int j;
float n,x0,y0,z0,k1,k2,k3,k4,xf,l1,l2,l3,l4,h,y,z;

cout<<"give how many interval u involved"<<endl;
cin>>n;
cout<<"give the value of initial condition x0,y0 and z0"<<endl;
cin>>x0>>y0>>z0;
cout<<"for which x value I want the value of y???"<<endl;
cin>>xf;

h=(xf-x0)/n;
//**************************
for(j=0;j<n;j++)
{
k1=h*gx(x0,y0,z0);
l1=h*fx(x0,y0,z0);
k2=h*gx((x0+(h/2.0)),(y0+(k1/2.0)),(z0+(l1/2.0)));
l2=h*fx((x0+(h/2.0)),(y0+(k1/2.0)),(z0+(l1/2.0)));
k3=h*gx((x0+(h/2.0)),(y0+(k2/2.0)),(z0+(l2/2.0)));
l3=h*fx((x0+(h/2)),(y0+(k2/2)),(z0+(l2/2)));
k4=h*gx((x0+h),(y0+k3),(z0+l3));
l4=h*fx((x0+h),(y0+k3),(z0+l3));
y=y0+(1.0/6.0)*(k1+2.0*k2+2.0*k3+k4);
z=z0+(1.0/6.0)*(l1+2.0*l2+2.0*l3+l4);
x0=x0+h;
y0=y;
z0=z;
out1<<x0<<"   "<<y<<"    "<<endl;
out2<<x0<<"   "<<z<<"   "<<endl;
}
}
```

Results:

```
toton@toton-VirtualBox ~/trial2/diff/runge_2nd $ g++ diff_q3.C -o diff_q3.x
toton@toton-VirtualBox ~/trial2/diff/runge_2nd $ ./diff_q3.x
give how many interval u involved
100
give the value of initial condition x0,y0 and z0
0
1
0
for which x value I want the value of y???
10
gnuplot> plot "diff_q3a.out"
gnuplot> replot "diff_q3b.out"
gnuplot> set grid
gnuplot> replot
```



*************INTERPOLATION*************

*********Inperpolation for Equal interval:************

Newton’s Forward Difference interpolation formula:

```
#include<iostream>
#include<cmath>
```

```
#include<fstream>
using namespace std;

int main(void)
{
int i,j,n=0;
float x[100],y[100][100],h,xnew,ynew,p;
ifstream in("new_for.in");
ofstream out("new_for.out");
//********************
while(in>>x[n]>>y[0][n])
n++;

cout<<"No of data point n = "<<n<<endl;
for(i=0;i<n;i++)
cout<<x[i]<<"    "<<y[0][i]<<endl;
//*******************
cout<<"Calculate delta(Yo):"<<endl;
for(i=1;i<n;i++)
{
        for(j=0;j<n-i;j++)
        {
        y[i][j]=y[i-1][j+1]-y[i-1][j];
        cout<<y[i][j]<<"\t";
        }cout<<endl;
}
//*********************
cout<<"Enter the value of x where we want to calculate y: ";
cin>>xnew;

h=x[1]-x[0];
cout<<"The equal spacing h = "<<h<<endl;
p=(xnew-x[0])/h; cout<<"P = "<<p<<endl;
//********************
float term=1.0,chenge;
ynew=y[0][0];
for(i=0;i<n-1;i++)
{
term=term*(p-i)/(i+1);
chenge=term*y[i+1][0];
ynew = ynew + chenge;
}
cout<<"xnew = "<<xnew<<"\t ynew = "<<ynew<<endl;

return 0;
}
```

Results:

```
toton@toton-VirtualBox ~/trial/interpolation $ g++ new_for.C -o new_for.x
toton@toton-VirtualBox ~/trial/interpolation $ ./new_for.x
No of data point n = 4
0    1
1    2
```

```
2    1
3    10
Calculate delta(Yo):
1         -1        9
-2        10
12
Enter the value of x where we want to calculate y: 4
The equal spacing h = 1
P = 4
xnew = 4         ynew = 41
```

Anather Results:

```
toton@toton-VirtualBox ~/trial/interpolation $ g++ new_forward.C -o new_forward.x
toton@toton-VirtualBox ~/trial/interpolation $ ./new_forward.x
No of data point n = 5
1950    151326
1960    179323
1970    203302
1980    226542
1990    249633
Calculate delta(Yo):
27997  23979  23240  23091
-4018  -739    -149
3279    590
-2689
Enter the value of x where we want to calculate y: 2000
The equal spacing h = 10
P = 5
xnew = 2000    ynew = 270476 (correct ans=281422)
```

*********Inperpolation for Unequal interval:************

@@@@@@@@@LAGRANGE INTERPOLATION @@@@@@@@@@

```
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;

int main(void)
{
int i,j,n=0;
float x[100],y[100],h,xnew,ynew,p;
ifstream in("lagrange.in");
//********************
while(in>>x[n]>>y[n])
n++;

cout<<"No of data point n = "<<n<<endl;
for(i=0;i<n;i++)
cout<<x[i]<<"    "<<y[i]<<endl;
//*********************
cout<<"Enter the value of x where we want to calculate y: ";
cin>>xnew;
//********************
float term,chenge;
ynew=0.0;
for(i=0;i<n;i++)
{
term=1.0;
            for(j=0;j<n;j++)
            {
            if(i!=j)
            term=term*((xnew-x[j])/(x[i]-x[j]));
            }
chenge=term*y[i];
ynew = ynew + chenge;
}
cout<<"xnew = "<<xnew<<"\t ynew = "<<ynew<<endl;

return 0;
}
```

Result:

```
toton@toton-VirtualBox ~/trial/interpolation $ g++ -o Wall lagrange.C -o lagrange.x
toton@toton-VirtualBox ~/trial/interpolation $ ./lagrange.x
No of data point n = 5
5    150
7    392
11    1452
13    2366
17    5202
Enter the value of x where we want to calculate y: 9
xnew = 9          ynew = 810
```

@@@@@@@@LOWER TRINGAL METHOD@@@@@@@@@@

***********SOLVING OF N NOF UNKOWN PARAMETER********

```
#include<iostream>
#include<cmath>
#include<fstream>
//.......................
using namespace std;
//.......................
int main()
{
float b[100],a[100][100],x[100];
int i,j,n=0,k;
ifstream ina("a.in");
ifstream inb("b.in");
//........Input & output  B.............
while(inb>>b[n])
n++;

cout<<"n=  "<<n<<"\nB==========="<<endl;
for(i=0;i<n;i++)
cout<<"\t"<<b[i]<<endl;
//........Input & output  B.............
cout<<"A=============="<<endl;
for(i=0;i<n;i++)
{
        for(j=0;j<n;j++)
        {
        ina>>a[i][j];
        cout<<a[i][j]<<"\t";
        }cout<<endl;
}
//..........Lower tringle making.................
for(j=0;j<=n;++j)
{
        for(i=j+1;i<=n;++i)
        {
        float m=a[i][j]/a[j][j]; //scale factor m
        b[i]-=m*b[j];

        for(k=j;k<=n;++k)
        a[i][k]=a[i][k]-m*a[j][k];
        }
}

cout<<"Lower tringle of A=============="<<endl;
for(i=0;i<n;i++)
{
        for(j=0;j<n;j++)
        {
        cout<<a[i][j]<<"\t";
        }cout<<endl;
}

//............For solution..................
for(i=n-1;i>=0;i--)
{
x[i]=b[i]/a[i][i];
        for(j=i+1;j<=n;++j)
        {
        x[i]=x[i]-( a[i][j]*x[j] )/a[i][i];
        }
}

cout<<"The solution of the equation:"<<endl;
for(i=0;i<n;i++)
cout<<"\t"<<x[i]<<endl;

return 0;
}
```

Results:

```
toton@sona ~/EXAM/prevoting $ g++ -O -Wall lowertringle.C -o lowertringle.x
toton@sona ~/EXAM/prevoting $ ./lowertringle.x
n=  3
B===========
            0.6867
            0.8338
            1
A==============
0.729   0.81     0.9
1           1           1
1.331   1.21     1.1
Lower tringle of A==============
0.729   0.81     0.9
0           -0.111111        -0.234568
0           0           0.0244441
The solution of the equation:
            0.224538
            0.281378
            0.327884
```

@@@@@@@@@**PREVOTING METHOD** @@@@@@@@@@

***********SOLVING OF N NOF UNKOWN PARAMETER********

```
#include<iostream>
#include<cmath>
#include<fstream>
//.......................
using namespace std;
//.......................
int main()
{
float b[100],a[100][100],x[100];
int i,j,n=0,k;
ifstream ina("a.in");
ifstream inb("b.in");
//........Input & output  B.............
while(inb>>b[n])
n++;

cout<<"n=  "<<n<<"\nB==========="<<endl;
for(i=0;i<n;i++)
cout<<"\t"<<b[i]<<endl;
//........Input & output  B.............
cout<<"A=============="<<endl;
for(i=0;i<n;i++)
{
            for(j=0;j<n;j++)
            {
            ina>>a[i][j];
            cout<<a[i][j]<<"\t";
            }cout<<endl;
}
//........If Diogonal element are zero...........
float amax,atemp,btemp;
int imax;
for(j=0;j<=n;++j)
{
amax=abs(a[j][j]);
imax=j;
	for(i=j+1;i<=n;++i)
	{
	if ( amax<abs(a[i][j]) )
		{
		amax=abs(a[i][j]);
		imax=i;
		}
	} //loop close i
	if(imax!=j)
	{
	btemp=b[j];
	b[j]=b[imax];
	b[imax]=btemp;
	for(k=j;k<=n;++k)
		{
		atemp=a[j][k];
		a[j][k]=a[imax][k];
		a[imax][k]=atemp;
		}
	}

}
//..............After prevoting out A & B.......
cout<<"After prevotin B=============="<<endl;
for(i=0;i<n;i++)
cout<<"\t"<<b[i]<<endl;

cout<<"After prevotin A=============="<<endl;
for(i=0;i<n;i++)
{
	for(j=0;j<n;j++)
	{
	cout<<a[i][j]<<"\t";
	}cout<<endl;
}

//..........Lower tringle making.................
for(j=0;j<=n;++j)
{
	for(i=j+1;i<=n;++i)
	{
	float m=a[i][j]/a[j][j]; //scale factor m
	b[i]-=m*b[j];

	for(k=j;k<=n;++k)
	a[i][k]-=m*a[j][k];
	}
}

cout<<"Lower tringle of A=============="<<endl;
for(i=0;i<n;i++)
```

```
{
        for(j=0;j<n;j++)
        {
        cout<<a[i][j]<<"\t";
        }cout<<endl;
}

//............For solution..................
for(i=n-1;i>=0;i--)
{
x[i]=b[i]/a[i][i];
        for(j=i+1;j<=n;++j)
        {
        x[i]=x[i]-( a[i][j]*x[j] )/a[i][i];
        }
}

cout<<"The solution of the equation:"<<endl;
for(i=0;i<n;i++)
cout<<"\t"<<x[i]<<endl;

return 0;
}
```

Results:

```
toton@sona ~/EXAM/prevoting $ g++ -O -Wall prevoting1.C -o prevoting1.X
toton@sona ~/EXAM/prevoting $ ./prevoting1.X
n=  3
B===========
        0.6867
        0.8338
        1
A==============
0.729   0.81    0.9
1       1       1
1.331   1.21    1.1
After prevotin B==============
        1
        0.8338
        0.6867
After prevotin A==============
1.331   1.21    1.1
1       1       1
0.729   0.81    0.9
Lower tringle of A==============
1.331   1.21    1.1
0       0.0909091       0.173554
5.96046e-08     0       0.0163635
The solution of the equation:
        0.224545
        0.281364
        0.327891
```