```
%PROGRAM FOR BISECTION METHOD
clc;
clear all;
f=input('Enter The Given Function f=');
n=input('Enter The Given Iritation n=');
y1=1;%For Initialising The While Loop
y2=1;%For Initialising The While Loop
while(y1*y2>0)
   a=input('Enter The Assumption a=');
   b=input('Enter The Assumption b=');
   y1=f(a);
   y2=f(b);
end;
for i=1:n
   x=(a+b)/2;
   y3=f(x);
   if y1*y3>0
      a=x;
      y1=y3;
   else
      b=x;
      y2=y3;
   end;
   fprintf('\n The Root Of The Given Equation f(x) Is %d',x);
end;
```

OUTPUT:

Enter The Given Function f=@(x)(-0.9*x^2+1.7*x+2.5)

Enter The Given Iritation n=6

Enter The Assumption a=2.5

Enter The Assumption b=3

The Root Of The Given Equation f(x) Is 2.750000e+00

The Root Of The Given Equation f(x) Is 2.875000e+00

The Root Of The Given Equation f(x) Is 2.812500e+00

The Root Of The Given Equation f(x) Is 2.843750e+00

The Root Of The Given Equation f(x) Is 2.859375e+00

The Root Of The Given Equation f(x) Is 2.867188e+00

SOLVER:

```
>> p=[-0.9 1.7 2.5];

>> r=roots(p)

r =  2.8601 -0.9712
```

%PROGRAM FOR NEWTONS RAPHSONS METHOD BY ACCURACY CRITERIA

```
clc;

clear all;

f=input('enter the function f=');

fd=input('enter the function fd=');
fdd=input('enter the function fdd=');
acc=input('enter the accuracy acc=');
c=2;%initialisation of the while loop
while(abs(c)>=1);
 x0=input('enter the assumption x0=');
c=(f(x0)*fdd(x0))/(fd(x0)^2);
end;
e=acc+1;
while(e>acc)
    x1=x0-(f(x0)/fd(x0));
    e=abs(x1-x0);
    x0=x1;
end;
fprintf('the root of the equation is %f',x1)
```

OUTPUT:

```
enter the function f=@(x)(cos(x)-x*exp(x))

enter the function fd=@(x)(-sin(x)-exp(x)*(x+1))

enter the function fdd=@(x)(-cos(x)-exp(x)*(x+2))

enter the accuracy acc=0.0001

enter the assumption x0=1

the root of the equation is 0.517757
```

SOLVER:

```
>> p=(cos(x)-x*exp(x));

>> roots=solve(p)

 roots =0.5177573636824582983227874741662 9
```

% PROGRAM FOR SUCCESSIVE APPROXIMATION METHOD

```
clc;
clear all;
f=input('enter the function f=');
ac=input('enter accuracy ac=');
c=1;%initialisation of while loop
while(c>0)
   a=input('enter a=');
   b=input('enter b=');
   c=f(a)*f(b);
end;
x0=(a+b)/2;
d=2; %intialization of while loop
while(d>=1)
   phi=input('enter the function phi=');
   phid=input('enter the function phid=');
   d=abs(phid(x0));
end;
e=ac+1;% initialisation of while loop
while(e>ac)
   x1=phi(x0);
   e=abs(x1-x0);
   x0=x1;
end;
fprintf('The root of the given function is %f',x1);
```

OUTPUT:

enter the function f=@(x)(x^2*exp(x)-1.4)

enter accuracy ac=0.001

enter a=0

enter b=1

enter the function phi=@(x)(sqrt(1.4)/exp(x/2))

enter the function phid=@(x)(-0.5916*exp(-x/2))

The root of the given function is 0.794891

SOLVER:

>> root= fzero(@(x) x^2*exp(x)-1.4,0.5)

root = 0.7951

```
%PROGRAM ON GAUSS-ELIMINATION METHOD
clc;
clear all;
a=input('Enter the partially pivoted matrix a=');
n=input('Enter the no. of unknowns n=');
for k=1:n-1
   for i=(k+1):n
      m=a(i,k)/a(k,k);
      for j=k:(n+1)
         a(i,j)=a(i,j)-m*a(k,j);
      end;
   end;
end;
disp(a);
for i=0:(n-1)
   s=0;
   for j=(n+1-i):n
      s=s+x(j)*a(n-i,j);
   end;
   x(n-i)=(a(n-i,n+1)-s)/a(n-i,n-i);
   fprintf('\n x(%d)=%f',n-i,x(n-i));
end;
```

OUTPUT:

Enter the partially pivoted matrix a=[3 2 1 7; 0 -1 1 1; 0 0 3 6]

Enter the no. of unknowns n=3

```
    3    2    1    7

    0   -1    1    1

    0    0    3    6
```

x(3)=2.000000

x(2)=1.000000

x(1)=1.000000

SOLVER:

```
a=[3 2 1; 0 -1 1; 0 0 3];
>> b=[7;1;6];
>> linsolve(a,b)
ans =

    1
    1
    2
```

```
%PROGRAM  FOR  GAUSS SIEDAL  METHODS
clc;
clear all;
u=input('Enter no. of unknowns u=');
n=input('Enter no. of iterations n=');
a=input('enter the coefficient matrix in partially pivoted form a=');
for i= 1:u
  x(i)=0;
end;
for k=1:n
  for i=1:u
    s=0;
    for j=1:u
      if i~=j
        s=s+a(i,j)*x(j);
      end;
    end;
    x(i)=(a(i,u+1)-s)/a(i,i);
  end;
end;
for i=1:u
  fprintf('\n The value of x(%d)=%f',i,x(i));
end;
```

OUTPUT:

enter the coefficient matrix=[10 1 1 12;2 10 1 13;2 2 10 14]

enter the no. of unknowns=3

enter the no. of itterations=5

the values are x(1)=1.200000

the values are x(2)=1.060000

the values are x(3)=0.948000

the values are x(1)=0.999200

the values are x(2)=1.005360

the values are x(3)=0.999088

the values are x(1)=0.999555

the values are x(2)=1.000180

the values are x(3)=1.000053

the values are x(1)=0.999977

the values are x(2)=0.999999

the values are x(3)=1.000005

the values are x(1)=1.000000

the values are x(2)=1.000000

the values are x(3)=1.000000

SOLVER:

```
>> a=[10 1 1;2 10 1;2 2 10];

>> b=[12;13;14];

>> linsolve(a,b)

ans =

    1.0000

    1.0000

    1.0000
```

%PROGRAM FOR R.K.2ND ORDER for simultaneous equation

```
clc;
clear all;
f=input('enter the Given function dy/dx f=');
g=input('enter the Given function d2y/dx2 g=');
x(1)=input('enter the value of x0=');
y(1)=input('enter the value of y0=');
z(1)=input('enter the value of z0=');
xn=input('enter the value of xn=');
h=input('enter the value of h=');
n=(xn-x(1))/h;
for i=1:n
  k1=h*f(x(i),y(i),z(i));
  l1=h*g(x(i),y(i),z(i));
  k2=h*f(x(i)+h,y(i)+k1,z(i)+l1);
  l2=h*g(x(i)+h,y(i)+k1,z(i)+l1);
  delk=(k1+k2)/2;
  dell=(l1+l2)/2;
  x(i+1)=x(i)+h;
  y(i+1)=y(i)+delk;
  z(i+1)=z(i)+dell;
end;
fprintf('/n y(%d)=%f,z(%d)=%f',(i+1),y(i+1),(i+1),z(i+1));
```

OUTPUT:

enter the Given function dy/dx f=@(x,y,z)(z)

enter the Given function d2y/dx2 g=@(x,y,z)(x*z^2-y^2)

enter the value of x0=0

enter the value of y0=1

enter the value of z0=0

enter the value of xn=0.2

enter the value of h=0.2

/n y(2)=0.980000,z(2)=-0.199200

% PROGRAM FOR STRAIGHT LINE CURVE FITTING y=ab^x

```
clc;
clear all;
x=input('Enter all values of x=');
y=input('Enter all values of y=');
n=input('Enter total no. of data points n=');
sx=0;
sy=0;
sx2=0;
sxy=0;
for i=1:n
  sx=sx+x(i);
  Y(i)=log(y(i));
  sy=sy+Y(i);
  sx2=sx2+(x(i)*x(i));
  sxy=sxy+(x(i)*Y(i));
end;
B=((sx*sy)-(n*sxy))/((sx*sx)-(n*sx2));
A=(sy-(B*sx))/n;
a=exp(A);
b=exp(B);
fprintf('Best fitting curve is y=%f*%f^x',a,b);
```

OUTPUT:

Enter all values of x=[1 2 3 4]

Enter all values of y=[4 11 35 100]

Enter total no. of data points n=4

Best fitting curve is y=1.326650*2.948829

SOLVER:

```
>>x=[1 2 3 4];

>> y=[4 11 35 100];

>> polyfit(x,y,1)

ans = 31.2000  -40.5000
```

% PROGRAM FOR STRAIGHT LINE CURVE FITTING y=a+bx

```
clc;
clear all;
x=input('Enter all values of x=');
y=input('Enter all values of y=');
n=input('Enter total no. of data points n=');
sx=0;
```

```
sy=0;
sx2=0;
sxy=0;
for i=1:n
  sx=sx+x(i);
  sy=sy+y(i);
  sx2=sx2+(x(i)*x(i));
  sxy=sxy+(x(i)*y(i));
end;
b=((sx*sy)-(n*sxy))/((sx*sx)-(n*sx2));
a=(sy-(b*sx))/n;
fprintf('Best fitting curve is y=%f+%fx',a,b);
```

OUTPUT:

Enter all values of x=[1 2 3 4 5 6 7]

Enter all values of y=[0.5 2.5 2 4 3.5 6 5.5]

Enter total no. of data points n=7

Best fitting curve is y=0.071429+0.839286x

SOLVER:

```
>> x=[1 2 3 4 5 6 7];

>> y=[0.5 2.5 2 4 3.5 6 5.5];

>> polyfit(x,y,1)

ans =

  0.8393    0.0714
```

% PROGRAM FOR STRAIGHT LINE CURVE FITTING y=a*x^b

```
clc;
clear all;
x=input('Enter all values of x=');
y=input('Enter all values of y=');
n=input('Enter total no. of data points n=');
sx=0;
sy=0;
sx2=0;
sxy=0;
for i=1:n
  X(i)=log(x(i));
  sx=sx+X(i);
  Y(i)=log(y(i));
  sy=sy+Y(i);
  sx2=sx2+(X(i)*X(i));
  sxy=sxy+(X(i)*Y(i));
end;
b=((sx*sy)-(n*sxy))/((sx*sx)-(n*sx2));
A=(sy-(b*sx))/n;
a=exp(A);
fprintf('Best fitting curve is y=%f*x^%f',a,b);
```

OUTPUT:

Enter all values of x=[1 2 3 4 5]

Enter all values of y=[0.5 2 4.5 8 12.5]

Enter total no. of data points n=5

Best fitting curve is y=0.500000*x^2.000000

SOLVER:

```
>> x=[1 2 3 4 5];

>> y=[0.5 2 4.5 8 12.5];

>> polyfit(x,y,2)

ans = 0.5000   -0.0000    0.0000
```

% PROGRAM FOR STRAIGHT LINE CURVE FITTING y=ae^bx

```
clc;
clear all;
x=input('Enter all values of x=');
y=input('Enter all values of y=');
n=input('Enter total no. of data points n=');
sx=0;
sy=0;
sx2=0;
sxy=0;
for i=1:n
  sx=sx+x(i);
  Y(i)=log(y(i));
  sy=sy+Y(i);
  sx2=sx2+(x(i)*x(i));
  sxy=sxy+(x(i)*Y(i));
end;
b=((sx*sy)-(n*sxy))/((sx*sx)-(n*sx2));
A=(sy-(b*sx))/n;
a=exp(A);
fprintf('Best fitting curve is y=%f*e^%fx',a,b);
```

OUTPUT:

Enter all values of x=[0 1 2 3]

Enter all values of y=[2 2.2103 2.4428 2.6997]

Enter total no. of data points n=4

Best fitting curve is y=1.999987*e^0.100000x

SOLVER:

```
>> x=[0 1 2 3];

>> y=[2 2.2103 2.4428 2.6997];

>> polyfit(x,y,1)

ans =  0.2332    1.9885
```

%PROGRAM FOR QUADRATIC CURVE FITTING

```
clc;
clear all;
x=input('enter the values of x=');
y=input('enter the values of y=');
n=input('enter the no. of data inputs n=');
sx=0;
sy=0;
sx2=0;
sx3=0;
sx4=0;
sxy=0;
sx2y=0;
for i=1:n
   sx=sx+x(i);
   sy=sy+y(i);
   sx2=sx2+x(i)*x(i);
   sxy=sxy+x(i)*y(i);
   sx3=sx3+x(i)*x(i)*x(i);
   sx4=sx4+x(i)*x(i)*x(i)*x(i);
   sx2y=sx2y+x(i)*x(i)*y(i);
end;
 p=[sx2 sx n; sx3 sx2 sx; sx4 sx3 sx2];
 q=[sy; sxy; sx2y];
 r=linsolve(p,q);

fprintf('Best fit curve is y=%fx^2+%f*x+c',r(1),r(2),r(3));
```

OUTPUT:

```
enter the values of x=[-3 -2 -1 0 1 2 3]
enter the values of y=[12 4 1 2 7 15 30]
enter the no. of data inputs n=7
Best fit curve is y=2.119048x^2+2.928571*x+cBest fit curve is y=1.666667x^2+
```

SOLVER:

```
x=[-3 -2 -1 0 1 2 3];
>> y=[12 4 1 2 7 15 30];
>> polyfit(x,y,2)
ans =    2.1190    2.9286    1.6667
```

```
% PROGRAM FOR LAGRANGES REVERSE INTERPOLATION
clc;
clear all;
x=input('enter the values of x=');
y=input('enter the values of y=');
yg=input('enter the values of yg=');
n=input('enter the total no. of data input=');
s=0;
for i=1:n
   p=1;
   for j=1:n
      if i~=j
         p=p*(yg-y(j))/(y(i)-y(j));
      end;
   end;
   L(i)=p*x(i);
   s=s+L(i);
end;
fprintf('yg=%f',s);
```

OUTPUT:

```
enter the values of x=[0 1 2 3]
enter the values of y=[0 1 7 25]
enter the values of yg=2
enter the total no. of data input=4
yg=1.716138
```

SOLVER:

```
>> y=[0 1 7 25];
>> x=[0 1 2 3];
>> xg=interp1(y,x,2)
xg =

   1.716138
```

```
%PROGRAM FOR NEWTON'S FORWARD INTERPOLATION
clc;
clear all;
x=input('Enter all the values of x=');
y=input('Enter all the values of y=');
n=input('Enter no. of data points n=');
xg=input('Enter value of x at which y is unknown xg=');
h=(x(2)-x(1));
p=(xg-x(1))/h;
for i=2:n
  for j=1:(n-i+1)
    y(i,j)=y((i-1),(j+1))-y((i-1),j);
  end;
end;
m=1;
yg=y(1,1);
for i=1
  m=m*(p-(j-1));%or p-j+1
  yg=yg+((m*y(1,(j+1)))/(factorial(j)));
end;
fprintf('The value of yg is yg=%f',yg);
```

OUTPUT:

```
Enter all the values of x=[0.1 0.3 0.5 0.7 0.9]
Enter all the values of y=[0.72 1.81 2.73 3.47 3.98]
Enter no. of data points n=5
Enter value of x at which y is unknown xg=0.2
The value of yg is yg=1.62500
```

SOLVER:

```
x=[0.1 0.3 0.5 0.7 0.9];
>> y=[0.72 1.81 2.73 3.47 3.98];
>> a=interp1(x,y,0.2)

a =1.2650
```

%PROGRAMM FOR LANGRANGEAN INTERPOLATION

```
clc;
clear all;
x=input('enter the values of x=');
y=input('enter the values of y=');
xg=input('enter the values of xg=');
n=input('enter the total no. of data input=');
s=0;
for i=1:n
   p=1;
   for j=1:n
      if i~=j
         p=p*(xg-x(j))/(x(i)-x(j));
      end;
   end;
   L(i)=p*y(i);
   s=s+L(i);
end;
fprintf('yg=%f',s);
```

OUTPUT:

enter the values of x=[0 1 2 5]

enter the values of y=[2 3 12 147]

enter the values of xg=1.5

enter the total no. of data input=4

yg=6.125000>>

SOLVER:

```
>> x=[0 1 2 5];

>> y=[2 3 12 147];

>> yg=interp1(x,y,1.5)


yg = 7.5000
```

% PROGRAM FOR TRAPEZOIDAL RULE

```
clc;
clear all;
f=input('enter the function f=');
xn=input('enter the value of xn=');
x0=input('enter the value of x0=');
n=input('enter the value of n=');
h=(xn-x0)/n;
for i=1:n
   x(i)=x0+(i-1)*h;
   y(i)=f(x(i));
end;
s=0;
for i=2:n-1
   s=s+y(i);
end;
A=(h/2)*(y(1)+y(n)+2*s);
fprintf('enter the valueof A=%f',A);
```

OUTPUT:

```
enter the function f=@(x)(1/(1+x^2))

enter the value of xn=1

enter the value of x0=0

enter the value of n=4

enter the valueof A=0.640294
```

SOLVER:

```
>> x=(0:0.25:1)

x =

        0    0.2500    0.5000    0.7500    1.0000

>> I=trapz(x,(1+x.^2).^-1)

I = 0.7828
```

% PROGRAM ON SIMPSONS 1/3rd OR 3/8th METHOD OF INTEGRATION

```
clc;
clear all;
f=input('enter the function f(x)=');
x0=input('enter the value of xo=');
xn=input('enter the value of xn=');
n=input('enter the value of steps n=');
h=(xn-x0)/n;
for i=1:n+1
  x(i)=x0+((i-1)*h);
  y(i)=f(x(i));
end;
c=input('Enter the type of integration for 1/3=1,for 3/8=2 c=');
 p=0;
    q=0;
if c==1
   for i=2:n
    if (mod(i,2)==0)
      p=p+y(i);
    else
      q=q+y(i);
    end;
  end;
 A=(h/3)*(y(1)+y(n+1)+4*p+2*q);
  fprintf('the value of integtartion is A=%f',A);
else
  for i=2:n
    if (mod(i-1,3)==0)
      p=p+y(i);
    else
      q=q+y(i);
    end;
  end;
  A=((3*h)/8)*(y(1)+y(n+1)+2*p+3*q);
fprintf('Te value of integration is A=%f',A);
end;
```

OUTPUT

```
enter the function f(x)=@(x)(exp(x))
enter the value of xo=0
enter the value of xn=4
enter the value of steps n=4
Enter the type of integration for 1/3=1,for 3/8=2 c=1
the value of integtartion is A=53.863846>>
```

enter the function f(x)=@(x)(exp(x))
enter the value of xo=0
enter the value of xn=4
enter the value of steps n=6
Enter the type of integration for 1/3=1,for 3/8=2 c=2
Te value of integration is A=53.717773>>

SOLVER

>> x=(0:1:4)

x =

    0   1   2   3   4

>> I=trapz(x,exp(x))

I = 53.9919

% PROGRAM FOR GQ2 &GQ3 METHOD

```
clc;
clear all;
f=input('enter the function f(x)=');
x0=input('enter the value x0=');
xn=input('enter the value xn=');
a=(xn-x0)/2;
b=(xn+x0)/2;
n=input('Enter the choice  for Gq2, 3 for GQ3 n=');
if n==2
  x1=a*(1/sqrt(3))+b;
  x2=a*(-1/sqrt(3))+b;
  y1=f(x1);
  y2=f(x2);
  A=a*(y1+y2);
  fprintf('The Value of integral is by GQ 2 is A=%f',A);
else
  x1=a*(sqtr(3/5))+b;
  x2=a*(0)+b;
  x3=a*(sqtr(-3/5))+b;
  y1=f(x1);
  y2=f(x2);
  y3=f(x3);
  A=a*(((5/9)*(y1+y3))+((8/9)*(y2)));
  fprintf('The value of integral by GQ3 is A=%f',A);
end;
```

OUTPUT:

enter the function f(x)=@(x)(x^3-cos(x)+6)

enter the value x0=0

enter the value xn=4

Enter the choice for Gq2, 3 for GQ3 n=2

The Value of integral is by GQ 2 is A=88.672814>>

SOLVER:

>> f='(x.^3-cos(x)+6)'

f =

(x.^3-cos(x)+6)

```
>> quad(f,0,4)

ans =   88.7568
```

%PROGRAM FOR PDE

```
clc;
clear all;
urt=input('enter the value at right side,urt=');
ult=input('enter the value at left side,ult=');
uup=input('enter the value at upper side,uup=');
udn=input('enter the value at bottom side,udn=');
a=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];
b=[uup+ult;uup+urt;ult+udn;udn+urt];
T=a\b;
fprintf('\nThe value of Temperature T=%f',T);
```

OUTPUT:

enter the value at right side,urt=0

enter the value at left side,ult=100

enter the value at upper side,uup=100

enter the value at bottom side,udn=0

The value of Temperature T=75.000000

The value of Temperature T=50.000000

The value of Temperature T=50.000000

The value of Temperature T=25.000000>>

Solver for maximization:

```
>> f=[-14 -20]

f =

  -14   -20

>> a=[20 6;40 8]

a =

   20     6

   40     8

>> b=[1000;500]

b =

       1000

        500

>> lb=[0,0];

>> [X,val]=linprog(f,a,b,[],[],lb)

Optimization terminated.

X =

    0.0000

   62.5000

val =

 -1.2500e+03
```

Solver For Minimization:

```
>> f=[25 30];

>> a=[4 3;2 3];

>> b=[60;36];

>> lb=[0,0];

>> [X,val]=linprog(f,a,b,[],[],lb)

Optimization terminated.

X =

   1.0e-10 *

    0.0900

    0.8809

val =

   2.8677e-09
```

% PROGRAM FOR DOUBLE INTEGRATION.

```
clc;
clear all;
f=input('enter the given function f(x)=');
x0=input('enter the value of x0=');
xn=input('enter the value of xn=');
y0=input('enter the value of y0=');
ym=input('enter the value of ym=');
n=input('enter the value of n=');
m=input('enter the value of m=');
h=(xn-x0)/n;
k=(ym-y0)/m;
for i=1:n+1
    x(i)=x0+(i-1)*h;
end;
for i=1:m+1
    y(i)=y0+(i-1)*k;
end;
for i=1:n+1
    for j=1:m+1
        p(i,j)=f(x(i),y(j));
    end;
end;
A1=0;
A2=0;
for i=2:m
    A1=A1+p(1,j)+p(n+1,1);
end;
for i=2:n
    A2=A2+p(i,1)+p(i,m+1);
end
A1=2*A1;
A2=2*A2;
A3=0;
for i=2:n
    for j=2:m
        A3=A3+p(i,j);
    end;
end;
A3=4*A3;
A4=p(1,1)+p(1,m+1)+p(n+1,1)+p(n+1,m+1);
A=(h*k/4)*(A1+A2+A3+A4);
fprintf('the value of area A=%f',A);
```

Output:

```
enter the given function f(x)=@(x,y) (x+y)

enter the value of x0=0

enter the value of xn=2

enter the value of y0=1
```

```
enter the value of ym=3

enter the value of n=2

enter the value of m=2

the value of area A=12.000000>>
```

Solver:

```
>> I=dblquad(@(x,y)x+y,0,2,1,3)

I =

    12
```

```
% Program for Euler's Method For Diffreential Equations
clc;
clear all;
f=input('enter the function f=');
x0=input('enter the value of x0=');
y0=input('enter the value of y0=');
h=input('enter the value of h=');
xn=input('enter the value of xn=');
n=(xn-x0)/h;
x(1)=x0;
y(1)=y0;
for i=2:n+1
  x(i)=x(i-1)+h;
  y(i)=y(i-1)+(h*f(x(i-1),y(i-1)));
end;
fprintf('value of y(%d)=%f',(n+1),y(n+1));
```

Output:

```
enter the function f=@(x,y) (x+y)

enter the value of x0=0

enter the value of y0=1

enter the value of h=0.25

enter the value of xn=1

value of y(5)=2.882813>>
```

Solver:

Save .m file with name EulER_SOlveR

```
function dy=f(x,y)
dy=(x+y);
```

@ command window:

```
>> [x,y]=ode45('EulER_SOlveR',[0:0.25:1],1)

x =

         0

    0.2500

    0.5000

    0.7500

    1.0000

y =

    1.0000

    1.3181
```

1.7974

2.4840

3.4366

```
%R.K. Method of 4th Order
clc;
clear all;
f=input('enter the function f=');
x0=input('enter the value of x0=');
y0=input('enter the value of y0=');
xn=input('enter the value of xn=');
h=input('enter the value of h=');
n=(xn-x0)/h;
x(1)=x0;
y(1)=y0;
for i=1:n+1
  x(i+1)=x(i)+h;
  k1=h*f(x(i),y(i));
  k2=h*f((x(i)+(h/2)),(y(i)+(k1/2)));
  k3=h*f((x(i)+(h/2)),(y(i)+(k2/2)));
  k4=h*f((x(i)+h),(y(i)+k3));
  dy=(k1+(2*k2)+(2*k3)+k4)/6;
  y(i+1)=y(i)+dy;
end;
fprintf('value of y(%d)=(%f)',(n+1),y(n+1));
```

Output:

```
enter the function f=@(x,y) ((x^2)+(y^2))
enter the value of x0=1
enter the value of y0=1.5
enter the value of xn=1.3
enter the value of h=0.1
value of y(4)=(3.554948)>>
```

Solver:

Save .m file with name R_k_4

```
function dy=f(x,y)
dy=((x^2)+(y^2));
```

@ command window:

```
>> [x,y]=ode45('R_k_4',[1:0.1:1.3],1.5)
x =
    1.0000
    1.1000
    1.2000
    1.3000
y =
    1.5000
    1.8955
    2.5044
```