clear all

close all

clc

zz=10;

while zz>1;

prompt='\n Please Enter the program to run: \n 1)Conversion of equation of Line from any form to other forms \n 2)Conversion of equation of Circle from any for to other forms \n 3)Checking Points of inersection/Tangency/No intersection between a Line and a Circle \n 4)Determining Points at distance from a point on a Line \n 5)Determining points at a distance(arcilength) from a point on a circle \n 6)Exit Program\n';

z=input(prompt);

if z==1;

%Choose number of lines to input

prompt='Input number of lines:';

n=input(prompt);

%Program to store data on each line input

i=0;

while i<(n)

%Choosing form of input

i=i+1;

fprintf('\nLine%d',i);

prompt='\n Please define the form of input: \n 1)Point and vector \n 2)Two Points \n 3)Implicit (ax+by+c=0) \n 4)Explicit (y=mx+c) \n';

j=input(prompt);

if j==1;%Point-Vector form

prompt='Input X coordinate of Point on Line:';

a(i)=input(prompt);

prompt='Input Y coordinate of Point on line:';

b(i)=input(prompt);

prompt='Input i component of Vector:';

u(i)=input(prompt);

prompt='Input j component of vector:';

v(i)=input(prompt);

fprintf 'Success!!'\n;

else if j==2;%Conversion of Two point to point and vector form

prompt='Input X coordinate of first Point on Line:';

a(i)=input(prompt);

prompt='Input Y coordinate of first Point on line:';

b(i)=input(prompt);

prompt='Input X coordinate of second Point on Line:';

d(i)=input(prompt);

prompt='Input Y coordinate of second Point on Line:';

e(i)=input(prompt);

if a(i)==d(i);%To avoid problem of infinite slope

u(i)=0;

v(i)=1;

fprintf 'Success!!\n';

else

u(i)=1;

v(i)=(e(i)-b(i))/(d(i)-a(i));

fprintf 'Success!!\n';

end

else if j==3;%Conversion of Implicit to Point and Vector form

prompt='Input value of a:';

f(i)=input(prompt);

prompt='Input value of b:';

g(i)=input(prompt);

prompt='Input value of c:';

h(i)=input(prompt);

if g(i)==0;%To avoid problem of infinite slope

a(i)=-h(i)/f(i);

b(i)=0;

u(i)=0;

v(i)=1;

fprintf 'Success!!\n';

else

a(i)=0;

b(i)=-h(i)/g(i);

u(i)=1;

v(i)=-f(i)/g(i);

fprintf 'Success!!\n';

end

else if j==4;%Conversion of Explicit to Point and Vector form

prompt='Input value of m:';

m(i)=input(prompt);

prompt='Input value of c:';

c(i)=input(prompt);

a(i)=0;

b(i)=c(i);

u(i)=1;

v(i)=m(i);

fprintf 'Success!!\n'

else i=i-1;

disp('INVALID INPUT\n')

end

end

end

end

end

for i=1:n;

%Obtaining other variables in terms of point and vector

m(i)=v(i)/u(i);

if u(i)==0;%Line parallel to Y-axis

if b(i)==0;%Line parallel to Y-axis and Point 1 is on X-axis

d(i)=a(i);

e(i)=1;

else%Line parallel to Y-axis and Point 1 not on X-axis

d(i)=a(i);

e(i)=0;

end

c(i)=inf;%Limitation of code. c=+or-inf, and can take infinite values if a(i)=0

f(i)=1;

g(i)=0;

h(i)=-a(i);

else%Line not parallel to Y-axis

c(i)=b(i)-m(i)\*a(i);

f(i)=m(i);

g(i)=-1;

h(i)=c(i);

if a(i)==0;%Given Point is on Y-axis

if b(i)==0;%Line passes through the origin

d(i)=u(i);

e(i)=v(i);

else%Line does not pass through the origin

d(i)=-b(i)/m(i);

e(i)=0;

end

else%Given point is not on Y-axis

d(i)=0;

e(i)=c(i);

end

end

end

for i=1:n;

fprintf('\n Line %d: \n Vector,Point:[%di+%dj],(%d,%d) \n Two point:(%d,%d),(%d,%d) \n Implicit: %dx+(%d)y+(%d)=0 \n Explicit: y=(%d)x+(%d)\n',i,u(i),v(i),a(i),b(i),a(i),b(i),d(i),e(i),f(i),g(i),h(i),m(i),c(i));

end

else if z==2;

%Choose number of circles to input

prompt='Input number of circles:';

n=input(prompt);

%Program to store data on each circle input

i=0;

while i<n;

i=i+1;

%Choosing form of input

prompt='\n Please define the form of input: \n 1)Centre and Radius \n 2)Implicit (ax^2+ay^2+bx+cy+d=0) \n 3)Explicit (y=sqrt[-x^2+ax+by]+c) \n';

j=input(prompt);

if j==1;

prompt='input X-coordinate of centre:';

o(i)=input(prompt);

prompt='input Y-coordinate of centre:';

s(i)=input(prompt);

prompt='input radius of circle:';

r(i)=input(prompt);

fprintf 'Success!!\n';

else if j==2;%Conversion of implicit to centre radius

prompt='input value of a:';

ca=input(prompt);

prompt='input value of b:';

cb=input(prompt);

prompt='input value of c:';

cc=input(prompt);

prompt='input value of d:';

cd=input(prompt);

o(i)=-cb/(2\*ca);

s(i)=-cc/(2\*ca);

r(i)=sqrt(-(cd/ca)+(o(i)^2)+(s(i)^2));

fprintf 'Success!!\n';

else if j==3;%conversion of explicit to centre radius

prompt='input value of a:';

ca=input(prompt);

prompt='input value of b:';

cb=input(prompt);

prompt='input value of c:';

cc=input(prompt);

s(i)=cc;

o(i)=ca/2;

r(i)=sqrt(cb+(o(i)^2));

fprintf 'Success!!\n';

else i=i-1;

fprintf 'INVALID INPUT\n'

end

end

end

end

for i=1:n;

fprintf('\n Circle %d: \n Parametric:x=%dcos(Theeta)+%d,y=%dsin(Theeta)+%d \n Implicit: (x-%d)^2+(y-%d)^2-(%d)^2=0 \n Explicit: y=sqrt[%d^2-(x-%d)^2]-%d\n',i,r(i),o(i),r(i),s(i),o(i),s(i),r(i),r(i),o(i),s(i));

end

else if z==3;

%Obtaining and reading Line:

i=0;

while i<1;

prompt='\n Please define the form of input for Line: \n 1)Point and vector \n 2)Two Points \n 3)Implicit (ax+by+c=0) \n 4)Explicit (y=mx+c) \n 5+)STOP PROGRAM \n';

j=input(prompt);

if j==1;%Point-Vector form

prompt='Input X coordinate of Point on Line:';

a=input(prompt);

prompt='Input Y coordinate of Point on line:';

b=input(prompt);

prompt='Input i component of Vector:';

u=input(prompt);

prompt='Input j component of vector:';

v=input(prompt);

i=2;

fprintf 'Success!!'\n;

else if j==2;%Conversion of Two point to point and vector form

prompt='Input X coordinate of first Point on Line:';

a=input(prompt);

prompt='Input Y coordinate of first Point on line:';

b=input(prompt);

prompt='Input X coordinate of second Point on Line:';

d=input(prompt);

prompt='Input Y coordinate of second Point on Line:';

e=input(prompt);

if a==d;%To avoid problem of infinite slope

u=0;

v=1;

i=2;

fprintf 'Success!!\n';

else

u=1;

v=(e-b)/(d-a);

i=2;

fprintf 'Success!!\n';

end

else if j==3;%Conversion of Implicit to Point and Vector form

prompt='Input value of a:';

f=input(prompt);

prompt='Input value of b:';

g=input(prompt);

prompt='Input value of c:';

h=input(prompt);

if g==0;%To avoid problem of infinite slope

a=-h/f;

b=0;

u=0;

v=1;

i=2;

fprintf 'Success!!\n';

else

a=0;

b=-h/g;

u=1;

v=-f/g;

i=2;

fprintf 'Success!!\n';

end

else if j==4;%Conversion of Explicit to Point and Vector form

prompt='Input value of m:';

m=input(prompt);

prompt='Input value of c:';

c=input(prompt);

a=0;

b=c;

u=1;

v=m;

i=2;

fprintf 'Success!!\n'

else i=0;

fprintf '\nINVALID INPUT\n'

end

end

end

end

end

%Obtaining other variables in terms of point and vector

m=v/u;

if u==0;%Line parallel to Y-axis

if b==0;%Line parallel to Y-axis and Point 1 is on X-axis

d=a;

e=1;

else%Line parallel to Y-axis and Point 1 not on X-axis

d=a;

e=0;

end

c=inf;%Limitation of code. c=+or-inf, and can take infinite values if a(i)=0

f=1;

g=0;

h=-a;

else%Line not parallel to Y-axis

c=b-m\*a;

f=m;

g=-1;

h=c;

if a==0;%Given Point is on Y-axis

if b==0;%Line passes through the origin

d=u;

e=v;

else%Line does not pass through the origin

d=-b/m;

e=0;

end

else%Given point is not on Y-axis

d=0;

e=c;

end

end

%Obtaining and reading Equation of Circle

i=0;

while i<1;

prompt='\n Please define the form of input for Circle: \n 1)Centre and Radius \n 2)Implicit (ax^2+ay^2+bx+cy+d=0) \n 3)Explicit (y=sqrt[-x^2+ax+by]+c) \n';

j=input(prompt);

if j==1;

prompt='input X-coordinate of centre:';

o=input(prompt);

prompt='input Y-coordinate of centre:';

s=input(prompt);

prompt='input radius of circle:';

r=input(prompt);

i=2;

fprintf 'Success!!\n';

else if j==2;%Conversion of implicit to centre radius

prompt='input value of a:';

ca=input(prompt);

prompt='input value of b:';

cb=input(prompt);

prompt='input value of c:';

cc=input(prompt);

prompt='input value of d:';

cd=input(prompt);

o=-cb/(2\*ca);

s=-cc/(2\*ca);

r=sqrt(-(cd/ca)+(o^2)+(s^2));

i=2;

fprintf 'Success!!\n';

else if j==3;%conversion of explicit to centre radius

prompt='input value of a:';

ca=input(prompt);

prompt='input value of b:';

cb=input(prompt);

prompt='input value of c:';

cc=input(prompt);

s=cc;

o=ca/2;

r=sqrt(cb+(o^2));

i=2;

fprintf 'Success!!\n';

else i=0;

fprintf '\nINVALID INPUT\n';

end

end

end

end

syms p q;%Solving both equations simultaneously

[solx,soly]=solve((p-o)^2+(q-s)^2-r^2==0,f\*p+g\*q+h==0);

tf = isreal(solx);%To check for real points of intersection

if tf==1;

tf = isreal(soly);%To check for real points of intersection

if tf==1;

if length(solx)==1%To check for Tangency

x=eval (solx(1));

y=eval (soly(1));

fprintf ('The given line is tangent at Point T (%d,%d)',x,y);

else

fprintf 'The Points of Intersection are:\n';

x=eval (solx(1));

y=eval (soly(1));

fprintf ('\nPoint A:(%d,%d)',x,y);

x=eval (solx(2));

y=eval (soly(2));

fprintf ('\nPoint B:(%d,%d)',x,y);

end

else

fpintf 'The Line and Circle do not intersect\n'

end

else

fprintf 'The Line and Circle do not intersect\n'

end

else if z==4;

%Obtaining and reading Line:

i=0;

while i<1;

prompt='\n Please define the form of input for Line: \n 1)Point and vector \n 2)Two Points \n 3)Implicit (ax+by+c=0) \n 4)Explicit (y=mx+c) \n 5+)STOP PROGRAM \n';

j=input(prompt);

if j==1;%Point-Vector form

prompt='Input X coordinate of Point on Line:';

a=input(prompt);

prompt='Input Y coordinate of Point on line:';

b=input(prompt);

prompt='Input i component of Vector:';

u=input(prompt);

prompt='Input j component of vector:';

v=input(prompt);

i=2;

fprintf 'Success!!'\n;

else if j==2;%Conversion of Two point to point and vector form

prompt='Input X coordinate of first Point on Line:';

a=input(prompt);

prompt='Input Y coordinate of first Point on line:';

b=input(prompt);

prompt='Input X coordinate of second Point on Line:';

d=input(prompt);

prompt='Input Y coordinate of second Point on Line:';

e=input(prompt);

if a==d;%To avoid problem of infinite slope

u=0;

v=1;

i=2;

fprintf 'Success!!\n';

else

u=1;

v=(e-b)/(d-a);

i=2;

fprintf 'Success!!\n';

end

else if j==3;%Conversion of Implicit to Point and Vector form

prompt='Input value of a:';

f=input(prompt);

prompt='Input value of b:';

g=input(prompt);

prompt='Input value of c:';

h=input(prompt);

if g==0;%To avoid problem of infinite slope

a=-h/f;

b=0;

u=0;

v=1;

i=2;

fprintf 'Success!!\n';

else

a=0;

b=-h/g;

u=1;

v=-f/g;

i=2;

fprintf 'Success!!\n';

end

else if j==4;%Conversion of Explicit to Point and Vector form

prompt='Input value of m:';

m=input(prompt);

prompt='Input value of c:';

c=input(prompt);

a=0;

b=c;

u=1;

v=m;

i=2;

fprintf 'Success!!\n'

else i=0;

fprintf '\nINVALID INPUT\n'

end

end

end

end

end

%Obtaining other variables in terms of point and vector

m=v/u;

if u==0;%Line parallel to Y-axis

if b==0;%Line parallel to Y-axis and Point 1 is on X-axis

d=a;

e=1;

else%Line parallel to Y-axis and Point 1 not on X-axis

d=a;

e=0;

end

c=inf;%Limitation of code. c=+or-inf, and can take infinite values if a(i)=0

f=1;

g=0;

h=-a;

else%Line not parallel to Y-axis

c=b-m\*a;

f=m;

g=-1;

h=c;

if a==0;%Given Point is on Y-axis

if b==0;%Line passes through the origin

d=u;

e=v;

else%Line does not pass through the origin

d=-b/m;

e=0;

end

else%Given point is not on Y-axis

d=0;

e=c;

end

end

%Obtaining the user defined point on line

i=2;

while i>1;

prompt='Input X coordinate of Point on Line:';

pa=input(prompt);

prompt='Input Y coordinate of Point on line:';

pb=input(prompt);

if f\*pa+g\*pb+h==0;%to check that point lies on the line

prompt='Input distance from given Point on Line:';%obtain distance

pd=input(prompt);

syms p q;%Solving both equations simultaneously

[solx,soly]=solve(sqrt((p-pa)^2+(q-pb)^2)==pd,f\*p+g\*q+h==0);

fprintf ('\nThe Points at a distance=%d from point (%d,%d) on the line are:\n',pd,pa,pb);

x=eval (solx(1));

y=eval (soly(1));

fprintf ('\nPoint A:(%d,%d)',x,y);

x=eval (solx(2));

y=eval (soly(2));

fprintf ('\nPoint B:(%d,%d)\n',x,y);

i=-10;%Exit loop

else

fprintf ('\nPoint (%d,%d) does not lie on the given line. Please check input.\n',pa,pb);

end

end

else if z==5;

%Obtaining and reading Equation of Circle

i=0;

while i<1;

prompt='\n Please define the form of input for Circle: \n 1)Centre and Radius \n 2)Implicit (ax^2+ay^2+bx+cy+d=0) \n 3)Explicit (y=sqrt[-x^2+ax+by]+c) \n';

j=input(prompt);

if j==1;

prompt='input X-coordinate of centre:';

o=input(prompt);

prompt='input Y-coordinate of centre:';

s=input(prompt);

prompt='input radius of circle:';

r=input(prompt);

i=2;

fprintf 'Success!!\n';

else if j==2;%Conversion of implicit to centre radius

prompt='input value of a:';

ca=input(prompt);

prompt='input value of b:';

cb=input(prompt);

prompt='input value of c:';

cc=input(prompt);

prompt='input value of d:';

cd=input(prompt);

o=-cb/(2\*ca);

s=-cc/(2\*ca);

r=sqrt(-(cd/ca)+(o^2)+(s^2));

i=2;

fprintf 'Success!!\n';

else if j==3;%conversion of explicit to centre radius

prompt='input value of a:';

ca=input(prompt);

prompt='input value of b:';

cb=input(prompt);

prompt='input value of c:';

cc=input(prompt);

s=cc;

o=ca/2;

r=sqrt(cb+(o^2));

i=2;

fprintf 'Success!!\n';

else i=0;

fprintf '\nINVALID INPUT\n';

end

end

end

end

%Obtaining the user defined point on circle

i=2;

while i>1;

prompt='Input X coordinate of Point on Circle:';

ca=input(prompt);

prompt='Input Y coordinate of Point on Circle:';

cb=input(prompt);

if (ca-o)^2+(cb-s)^2==r^2;%to check that point lies on the circle

th=atan((cb-s)/(ca-o));

prompt='Input arc-length from the given Point on Circle:';%obtain arc-length

cd=input(prompt);

th1=(cd/r)+th;

if abs(cos(th1))==1;%for pi/2 and pi

fprintf ('\nThe Point at an arclength=%d from point (%d,%d) on the circle is:\n',cd,ca,cb);

x1=r\*cos(th1)+o;

y1=r\*sin(th1)+s;

fprintf ('\nPoint A:(%d,%d)',x1,y1);

i=-10;%Exit loop

else

fprintf ('\nThe Points at an arclength=%d from point (%d,%d) on the circle are:\n',cd,ca,cb);

x1=r\*cos(th1)+o;

x2=r\*cos(-th1)+o;

y1=r\*sin(th1)+s;

y2=r\*sin(-th1)+s;

fprintf ('\nPoint A:(%d,%d)',x1,y1);

fprintf ('\nPoint B:(%d,%d)\n',x2,y2);

i=-10;%Exit loop

end

else

fprintf ('\nPoint (%d,%d) does not lie on the given circle. Please check input.\n',ca,cb);

end

end

else if z==6;

zz=-10;

else

fprintf 'INVALID INPUT';

end

end

end

end

end

end

end