i=imread('C:\Users\20b-051-CS\Desktop\cc.jpg');

k = imresize(i,[1200 1500]);

j = imrotate(k,90);

l=rgb2gray(j);

m=im2bw(l);

subplot(1,3,1),imshow(i),title('Original')

subplot(1,3,2),imshow(k),title('Resized')

subplot(1,3,3),imshow(j),title('Rotated')

Task 1: Draw a line using Bresenham’ Algorithm, Consider line starting and ending points as (20, 10) & (30, 18)

Code:

x0=20; x1=30; y0= 10; y1= 18;

Dx= x1-x0; Dy=y1-y0;

pk= (2\*Dy)-Dx;

tic

a = 0;

while a<10

a=a+1;

if pk<0

x0=x0+1;

pk=pk+2\*Dy;

else

x0=x0+1;

y0=y0+1;

pk=pk+2\*Dy-2\*Dx;

end

hold on

plot (x0, y0, '\*')

end

toc

**Task 1: (Octate)**

**Code:**

r= input('enter radius: ');

x=0;

y=r;

P= 1-r;

a= 2\*x+2;

b=2\*y-2;

while a~=b

hold on

if P<0

x=x+1;

y=y;

P=P+2\*x+1;

else

x=x+1;

y=y-1;

P=P+2\*x-2\*y;

end

a= 2\*x+2;

b=2\*y-2;

plot (x,y,'+')

end

hold off

**Task 2: (Quad)**

**Code:**

r = input(' Enter the value for radius: ');

x = 0;

y = r;

ix = 2\*x+2;

iy = 2\*y-2;

i=0;

p = 1-r;

while y ~= 0;

hold on

plot (x,y,'+')

if p < 0

x = x+1;

y = y;

p = p+2\*x+1;

else

x = x+1;

y = y-1;

p = p+2\*x-2\*y;

end

ix = 2\*x+2;

iy = 2\*y-2;

end

hold off

**Task 3: (Circle)**

**Code:**

n = input('Enter no of points: ');

angle = 0:2\*pi/n:2\*pi;

r = 1;

x1 = r\*cos(angle);

y1 = r\*sin(angle);

x2 = r\*cos(angle+2\*pi/n);

y2 = r\*sin(angle+2\*pi/n);

line([x1 x2],[y1 y2]);

axis equal;

grid on

**Translation:**

factor = [2 , 3];

matA = [11 , 117];

matB = [68 , 18];

matC = [126 , 117];

matD = matA + factor;

matE = matB + factor;

matF = matC + factor;

disp(matD);

disp(matE);

disp(matF);

hold on;

plot(matD(:,1),matD(:,2),'\*');

plot(matE(:,1),matE(:,2),'\*');

plot(matF(:,1),matF(:,2),'\*');

legend('Points after translation')

hold off

hold on;

plot(matA(:,1),matA(:,2),'+');

plot(matB(:,1),matB(:,2),'+');

plot(matC(:,1),matC(:,2),'+');

hold off

Scaling:

matA = [11 ; 117];

matB = [68 ; 18];

matC = [126 ; 117];

factor = [10 , 0 ; 0 10];

matD = factor \* matA;

matE = factor \* matB;

matF = factor \* matC;

disp(matD);

disp(matE);

disp(matF);

hold on;

plot(matD(1,:),matD(2,:),'\*');

plot(matE(1,:),matE(2,:),'\*');

plot(matF(1,:),matF(2,:),'\*');

legend('Points after Scaling')

hold off

hold on;

plot(matA(1,:),matA(2,:),'+');

plot(matB(1,:),matB(2,:),'+');

plot(matC(1,:),matC(2,:),'+');

hold off

Rotation:

x = 45;

factor = [cos(x) sin(x) 0; -sin(x) cos(x) 0; 0, 0, 1];

matA = [11 ,117, 12];

matB = [68 ,18, 65];

matC = [126 ,117, 120];

matD = matA \* factor;

matE = matB \* factor;

matF = matC \* factor;

disp(matD);

disp(matE);

disp(matF);

hold on;

plot3(matF(:,1),matF(:,2),matF(:,3),'\*');

plot3(matE(:,1),matE(:,2),matE(:,3),'\*');

plot3(matD(:,1),matD(:,2),matD(:,3),'\*');

legend('Points after Rotation')

hold off

hold on;

plot(matA(:,1),matA(:,2),'+');

plot(matB(:,1),matB(:,2),'+');

plot(matC(:,1),matC(:,2),'+');

hold off

Bezier Curve Implementation

**Code:**

x1= 0;

y1=0;

x2=3;

y2=4;

x1x2 = (x1+x2)/2;

y1y2 = (y1+y2)/2;

mid1x = (x1+x1x2)/2;

mid1y = (y1+y1y2)/2;

x3=5;

y3=0;

x2x3 = (x2+x3)/2;

y2y3 = (y2+y3)/2;

mid3x = (x3+x2x3)/2;

mid3y = (y3+y2y3)/2;

mid2x = (x1x2+x2x3)/2;

mid2y = (y1y2+y2y3)/2;

hold on;

plot(x1,y1,'\*');

plot(x2,y2,'\*');

plot(x3,y3,'\*');

plot(x1x2,y1y2,'+');

plot(x2x3,y2y3,'+');

plot(mid1x,mid1y,'+');

plot(mid2x,mid2y,'+');

plot(mid3x,mid3y,'+');

hold off;

**For Line plotting:**

plot ([x1 mid1x],[y1 mid1y]);

plot ([mid1x mid2x],[mid1y mid2y]);

plot ([mid2x mid3x],[mid2y mid3y]);

plot ([mid3x x3],[mid3y y3]);

**Filters**

**Task 1: (Sharpening and Blurring the image)**

Code:

i = imread('Pout.tif');

j = imsharpen(i, 'Radius',5, 'Amount',3);

b = fspecial('average',[10,10]);

g = imfilter(i,b);

subplot(1,3,1),imshow(i), title('Original')

subplot(1,3,2),imshow(j), title('Sharpen Image')

subplot(1,3,3),imshow(g), title('Blured Image')

**Task 2: (Edges)**

Code:

i = imread('C:\Users\20B-063-CS\Desktop\Leaf.jfif');

j = rgb2gray(i);

k = edge(j,'sobel');

l = edge(j,'canny');

subplot(1,3,1),imshow(j),title('Grayscale Image')

subplot(1,3,2),imshow(k),title('Sobel Effect')

subplot(1,3,3),imshow(l),title('Canny Effect')