**PROGRAM 1**

#include<graphics.h>

#include<conio.h>

#include<stdio.h>

#include<math.h>

Void main()

{

int gd=DETECT,gm;

int dx,dy,p,end;

float x1,x2,y1,y2,x,y;

initgraph(&gd,&gm,”c:\\tc\\bgi”);

printf(“\n\t\t BRESENHAMS LINE DRAWING ALGORITHM”);

printf(“\n Enter the value of x1&X2:”);

scanf(“%f%f”,&x1,&x2);

printf(“\n Enter the value of y1&y2:”);

scanf(“%f%f”,&y1,&y2);

dx=abs(x1-x2);

dy=abs(y1-y2);

p=2\*dy-dx;

if(x1>x2)

{

x=x2;

y=y2;

end=x1;

}

else

{

x=x1;

y=y1;

end=x2;

}

putpixel(x,y,10);

while(x<end)

{

x=x+1;

if(p<0)

{

p=p+2\*dy;

}

else

{

y=y+1;

p=p+2\*(dy-dx);

}

putpixel(x,y,10);

}

getch();

closegraph();

}

**PROGRAM 2**

#include<graphics.h>

#include<conio.h>

#include<stdio.h>

#include<math.h>

int gd=DETECT,gm=0,mx,my,i;

float x[20],y[20];

linedr()

{

int i;

mx=getmaxx()/2;

my=getmaxy()/2;

settextstyle(3,1,2);

outtextxy(304,1,”>”);

settextstyle(3,0,2);

outtextxy(327,1,“+y”);

outtextxy(314,450,“v”);

outtextxy(340,455,“-y”);

outtextxy(2,255,“<”);

outtextxy(0,250,“-x”);

outtextxy(628,255,“>”);

outtextxy(,610240,“+x”);

line(mx,10,mx,(my\*2)-10);

line(10,my,(mx\*2)-10,my);

getch();

return 0;

}

void main()

{

float j,te,sx,sy,xf,yf,s,p,tx,ty,xr=0,yr=0;

int c,n;

char cc=’y’;

clrscr();

initgraph(&gd,&gm,”c:\\tc\\bgi”);

printf(“\n\t\t BASIC TANSFORMATION”);

printf(“\n\t\t ………. …………………………..”);

printf(“\n\t\t Enter the co-ordinates of triange:”);

for(i=1;i<=4;i++);

scanf(“%f%f”,&x[i],&y[i]);

printf(“\nMenu\n1.Translation”);

printf(“\n2.Scaling”);

printf(“\n3.Rotating”);

printf(“\n Enter the choice:”);

scanf(“%d”,&c);

switch(c)

{

Case 1:

{

printf(“\n Enter the transformation distance(tx&ty):”);

scanf(“%f%f”,&tx,&ty);

cleardevice();

outtextxy(1,1,”Before Tranformation”);

linedr();

cleardevice();

for(i=1;i<=4;i++)

{

x[i]=x[i]+tx;

y[i]=y[i]+ty;

}

outtextxy(1,1,”After Tranformation”);

linedr();

break;

}

Case 2:

{

cleardevice();

printf(“\n Enter the scaling distance(sx&sy):”);

scanf(“%f%f”,&sx,&ty);

printf(“\n Enter the fixedpointvalue(xf&yf);

scanf(“%f%f”,&xf,&yf);

cleardevice();

outtextxy(1,1,”Before Scaling”);

linedr();

cleardevice();

for(i=1;i<=4;i++)

{

x[i]=x[i]\*sx+(1-sx)\*xf;

y[i]=y[i] ]\*yx+(1-yx)\*yf;

}

outtextxy(1,1,”After Scaling ”);

linedr();

break;

}

Case 3:

{  
printf(“\n Enter rotating angle(teta):”);

scanf(“%f”,&te);

printf(“\n Enter the fixedpoint(xr&yr):”);

scanf(“%f%f”,&xr,&yr);

cleardevice();

outtextxy(1,1,”Before rotation”);

linedr();

j=te\*3.14/180;

for(i=1;i<=4;i++)

{

s=cos(j);

p=sin(j);

x[i]=xr+(x[i]-xr)\*sx-( y[i]-yr)\*p;

y[i]=yr+(y[i]-yr)\*sy+(x[i]-xr)\*p;

}

cleardevice();

outtextxy(1,1,”After rotation” ”);

linedr();

}

}

Clrscr();

cleardevice();

printf(“\n Do you want to continue(y/n):”);

scanf(“%s”,<&cc);

while(cc==’y’);

}

**PROGRAM-4**

#include <graphics.h>

#include<conio.h>

#include <math.h>

#include<stdio.h>

#define pi 3.14

float startangle, endangle:

int x,y;

int can\_draw(float teta)

{

if (teta>=startangle && teta<=endangle)

return 1;

return 0;

}

void circlepoint (int x, int y, int xc, int yc)

float teta;

teta= atan ((float)y/x);

/ /teta-teta(180/pi);

if (can\_draw(teta))

putpixel(xc+x, yc-y,WHITE);

if (can\_draw(360-teta))

putpixel(xc+x,yc+y,WHITE);

if (can\_draw(90-teta))

putpixel(xc+y,yc-x,WHITE);

if (can\_draw(270+teta))

putpixel(xc+y,yc+x,WHITE);

if (can\_draw(180+teta))

putpixel(xc-x,yc-y,WHITE);

if (can\_draw(180+teta))

putpixel(xc-x, yc+y,WHITE);

if (can\_draw(90+teta))

putpixel(xc-y,yc-x,WHITE);

if (can\_draw(270-teta))

putpixel(xc-y,yc+x,WHITE);

}

void midpointcircle (int xc, int yc, int rad)

float d=(5/4.0)-rad;

x=0;y=rad;

while(y>x)

{

if (d<0)

d+=2\*x+3;

else

d+=(2\*x)−(2\*y)+5,y--;

x++;

circlepoint (x,y,xc,yc);

}

}

void main()

int gd-DETECT,gm;

int radius,xc,yc, choice, temp;

float xstart,ystart,xend,yend;

initgraph(&gd, &gm, "c:\\tc\\bgi");

cleardevice();

printf("\t\t\t MIDPOINTCIRCLE");

printf("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf CNn Enter the center (x,y):");

scanf (“%d%d", &xc,&yc);

printf("\n Enter the radius: ");

scanf ("%d", &radius);

cleardevice();

settextstyle(1,0,2);

startangle=0,endangle=360;

midpointcircle(xc,yc, radius);

getch();

}