**Experiment 1: Scan Conversion of a point.**

**Source code:**

#include<bits/stdc++.h>

#include<graphics.h>

using namespace std;

int main()

{

int gd=DETECT,gm;

initgraph(&gd, &gm, "c:\\TC\\BGI");

float x=300,y=500;

putpixel(x,y,GREEN);

getch();

closegraph();

return 0;

}

**Output Screenshot:**

****

**Experimen 2: Scan Converting a line using DDA Algorithm.**

**Source Code:**

#include<bits/stdc++.h>

#include<graphics.h>

int main()

{

int gd=DETECT,gm,i;

float x,y,dx,dy,step;

int x0,y0,y1,x1;

initgraph(&gd,&gm,"C:\\TC\\BGI");

setbkcolor(WHITE);

x0=100,y0=200,x1=500,y1=300;

dx=float(x1-x0);

dy=float(y1-y0);

if(dx>=dy) step=dx;

else step=dy;

dx=dx/step;

dy=dy/step;

x=x0,y=y0;

printf("%f\n",step);

i=1;

while(i<=step)

{

putpixel(x,y,RED);

x+=dx;

y+=dy; i++;

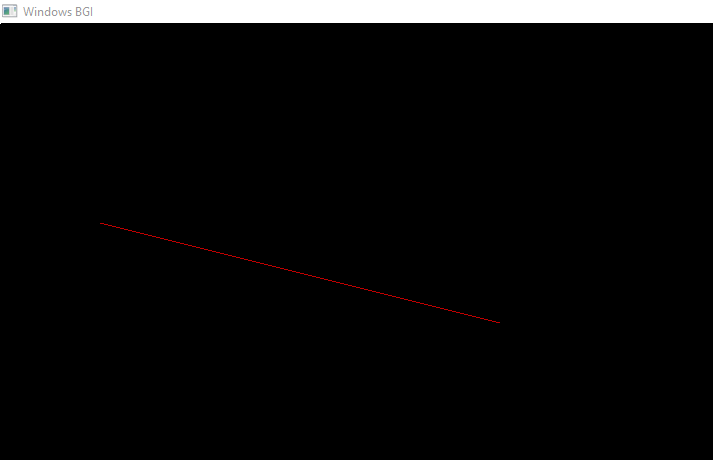
}

getch();

closegraph();

}

**Output Screenshot:**



**Experiment 3: Scan conversion a line using Bresenham’s Algorithm.**

**Source Code:**

#include<bits/stdc++.h>

#include<stdio.h>

#include<graphics.h>

void drawline(int x0, int y0, int x1, int y1)

{

int dx, dy, p, x, y;

dx=x1-x0;

dy=y1-y0;

x=x0;

y=y0;

p=2\*dy-dx;

while(x<x1)

{

if(p>=0)

{

putpixel(x,y,9);

y=y+1;

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

}

else

{

putpixel(x,y,9);

p=p+2\*dy;

}

x=x+1;

}

}

int main()

{

int gd=DETECT, gm, error, x0, y0, x1, y1;

initgraph(&gd, &gm,"c:\\turboc3\\bgi");

printf("Enter co-ordinates of first point: ");

scanf("%d%d", &x0, &y0);

printf("Enter co-ordinates of second point: ");

scanf("%d%d", &x1, &y1);

drawline(x0, y0, x1, y1);

getch();

return 0;

}

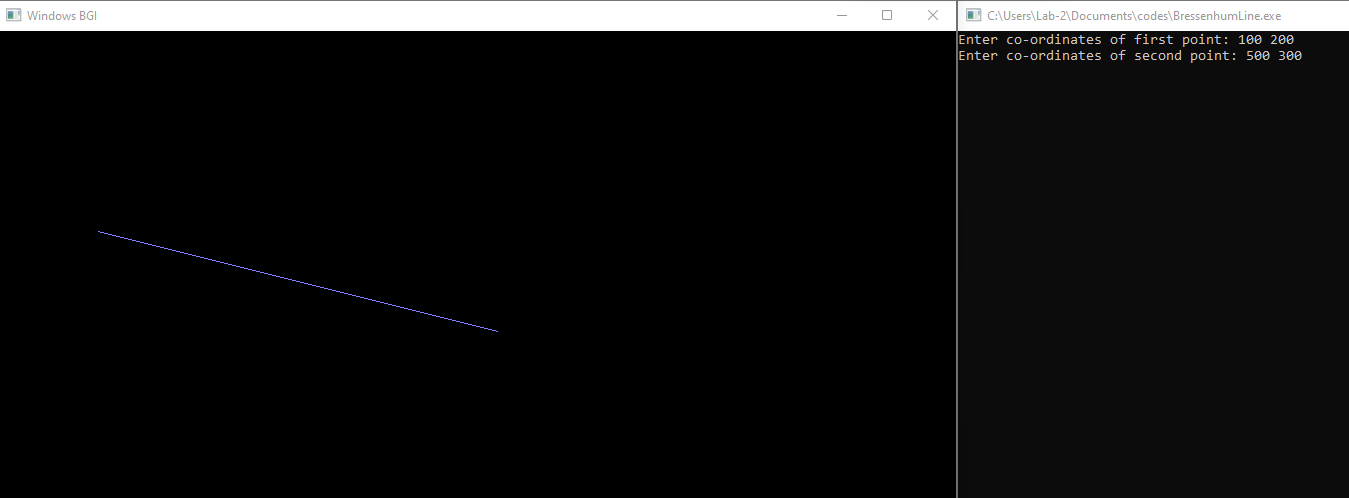
/\*

100 100

200 200

\*/

**Output Screenshot:**



**Experiment 4 : Scan converting a circle using Bresenham’s Algorithm.**

**Source Code:**

#include<bits/stdc++.h>

#include <graphics.h>

#include <stdlib.h>

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

{

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

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

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

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

putpixel(y+xc,x+yc,12);

putpixel(y+xc,-x+yc,14);

putpixel(-y+xc,-x+yc,15);

putpixel(-y+xc,x+yc,6);

}

void BresenhamCircle(int xc,int yc,int r)

{

int x=0,y=r,d=3-(2\*r);

EightWaySymmetricPlot(xc,yc,x,y);

while(x<=y)

{

if(d<=0)

d=d+(4\*x)+6;

else

{

d=d+(4\*x)-(4\*y)+10;

y=y-1;

}

x=x+1;

EightWaySymmetricPlot(xc,yc,x,y);

}

}

int main(void)

{

int xc,yc,r,gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

printf("Enter the values of xc and yc :");

scanf("%d%d",&xc,&yc);

printf("Enter the value of radius :");

scanf("%d",&r);

BresenhamCircle(xc,yc,r);

getch();

closegraph();

return 0;

}

**Output Screenshot:**

