CGM ASSIGNMENT

Nivedi Singhal

189301049

CSE-C

1. Draw a Circle using MidPoint circle algorithm and Bresenham’s Circle algorithm.

* MID POINT

#include<iostream> using namespace std;

void midPointCircleDraw(int x\_centre, int y\_centre, int r)

{

int x=r, y=0;

cout<< "("<< x+x\_centre << ", " << y+y\_centre << ") ";

if (r>0)

{

cout<<"("<<x+x\_centre << ", " <<-y+y\_centre<<") "; cout<<"("<<y+x\_centre << ", " <<x+y\_centre<<") "; cout<<"("<<-y+x\_centre << ", " <<x+y\_centre<<")\n";

}

int P=1-r; while (x > y)

{

y++;

if (P<=0) P=P+2\*y+1;

else

{

x--;

P=P+2\*y-2\*x+1;

}

if (x<y) break;

cout <<"("<<x+x\_centre << ", " <<y+y\_centre<<") "; cout <<"("<<-x+x\_centre << ", " <<y+y\_centre<<") "; cout <<"("<<x+x\_centre << ", " <<-y+y\_centre<<") "; cout <<"("<<-x+x\_centre << ", " <<-y+y\_centre<<")\n";

if (x!=y)

{

cout<<"("<< y+x\_centre << ", " <<x+y\_centre<<") "; cout<<"("<<-y+x\_centre << ", " <<x+y\_centre<<") "; cout<<"("<<y+x\_centre << ", " <<-x+y\_centre<<") "; cout<<"("<<-y+x\_centre << ", " <<-x+y\_centre<<")\n";

}

}

}

int main()

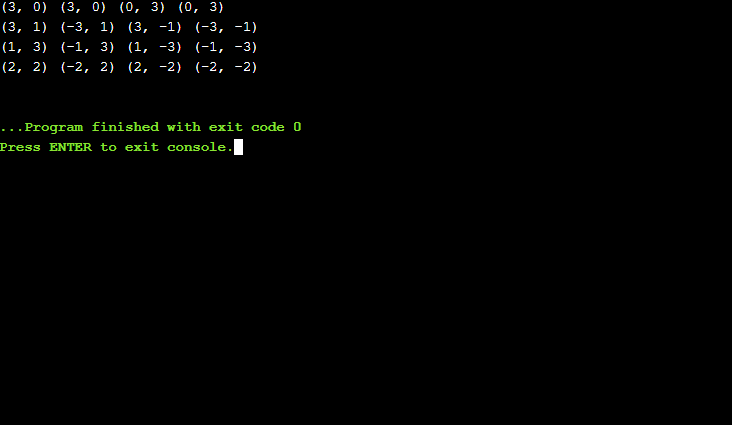
{

midPointCircleDraw(0, 0, 3);

return 0;

}

**OUTPUT**



BRESENHAM CIRCLE ALGORITHM

#include<iostream.h> #include<graphics.h>

void drawcircle(int x0, int y0, int r)

{

int x=r; int y=0; int err=0;

int main()

{

int gdriver=DETECT, gmode, error, x, y, r; initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");

cout<<"Enter radius of circle: "; cin>>r;

cout<<"Enter co-ordinates of center(x and y): "; cin>>x>>y;

drawcircle(x, y, r);

return 0;

}

1. Draw an Ellipse for Region 1 and Region 2 using Midpoint Ellipse Algorithm