Lab – 3

5. Midpoint line drawing.

#include <stdio.h>

#include <graphics.h>

#include <math.h>

int main()

{

    int gd = DETECT, gm, color;

    initgraph(&gd, &gm, "");

    int x0 = 30, y0 = 50;

    int x1 = 400, y1 = 100;

    int dx = x1 - x0;

    int dy = y1 - y0;

    int yk = y0;

    int xk = x0;

    float m = (float)abs(dy) / (float)abs(dx);

    if (m < 1.0)

    {

        int pk = abs(dy) - (abs(dx) / 2);

        while (xk != x1 || yk != y1)

        {

            xk++;

            if (pk < 0)

            {

                pk += dy;

            }

            else

            {

                pk += dy - dx;

                yk++;

            }

            putpixel(xk, yk, RED);

        }

    }

    else

    {

        int pk = abs(dx) - (abs(dy) / 2);

        while (xk != x1 || yk != y1)

        {

            yk++;

            if (pk < 0)

            {

                pk += dx;

            }

            else

            {

                pk += dx - dy;

                xk++;

            }

            putpixel(xk, yk, RED);

        }

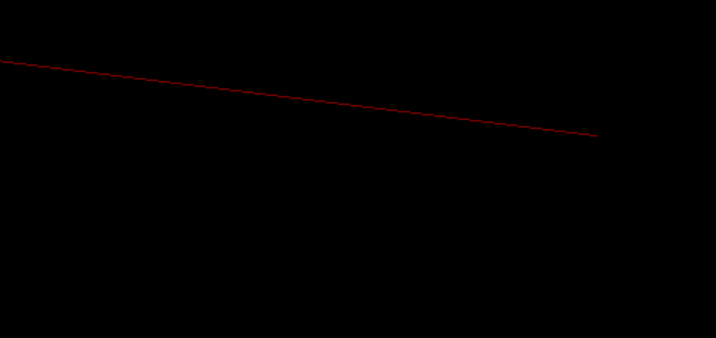
    }

    getch();

    closegraph();

    return 0;

}



6. ⁠midpoint circle drawing.

#include <stdio.h>

#include <graphics.h>

#include <math.h>

int main()

{

    int gd = DETECT, gm;

    initgraph(&gd, &gm, "");

    int xc = 350;

    int yc = 140;

    int r = 80;

    int yk = r;

    int xk = 0;

    int pk = 1 - r;

    while (xk <= yk)

    {

        xk++;

        if (pk < 0)

        {

            pk += 2 \* xk + 3;

        }

        else

        {

            yk--;

            pk += 2 \* xk - 2 \* yk + 5;

        }

        putpixel(xc + xk, yc + yk, GREEN);

        putpixel(xc + yk, yc + xk, RED);

        putpixel(xc - xk, yc + yk, GREEN);

        putpixel(xc - yk, yc + xk, RED);

        putpixel(xc + xk, yc - yk, GREEN);

        putpixel(xc + yk, yc - xk, RED);

        putpixel(xc - xk, yc - yk, GREEN);

        putpixel(xc - yk, yc - xk, RED);

    }

    getch();

    closegraph();

    return 0;

}

