



```

static void DemoThree(PDF p)
{
    // Create an image, 500 pixels square

    int width = 500;
    int height = 500;

    var anImage = new List<List<RGB>>();

    for(int i = 0; i < height; i ++)
    {
        var row = new List<RGB>();

        for(int j = 0; j < width; j ++)
            row.Add(new RGB(0, 0, 0));

        anImage.Add(row);
    }

    double yStart = -2.0;
    double yStop = 2.0;
    double yStep = (yStop - yStart) / (height - 1);

    double xStart = -2.0;
    double xStop = 2.0;
    double xStep = (xStop - xStart) / (width - 1);

    int maxIterations = 25;
    double maxDistance = 1000.0;

    int iValue = 0;

    for(double y = yStart; y <= yStop; y += yStep)
    {
        int jValue = 0;

        for(double x = xStart; x <= xStop; x += xStep)
        {
            Complex z = new Complex(0.0, 0.0);
            Complex c = new Complex(x, y);

            int iterations = 0;

            while(
                iterations < maxIterations &&
                Math.Sqrt(z.X * z.X + z.Y * z.Y) < maxDistance
            )
            {
                z = z.Multiply(z).Add(c);
                iterations ++;
            }

            double v1 = (double)iterations / maxIterations;
            double v2 = Math.Sqrt(v1);
            double v3 = Math.Sqrt(v2);

            v1 *= 255.0;
            v2 *= 255.0;
            v3 *= 255.0;

            byte red = (byte)Math.Round(v1);
            byte green = (byte)Math.Round(v2);
            byte blue = (byte)Math.Round(v3);

            var theColor = new RGB(red, green, blue);

            anImage[iValue][jValue] = theColor;

            //std::cout << iValue << " " << jValue << " ";

            jValue++;
        }

        iValue++;
    }

    // Place the image, centered

    ImageInfo info = p.ProcessImage(anImage);

    int xValue = (p.GetWidth() - width) / 2;
    int yValue = (p.GetHeight() - height) / 2;

    p.ShowImage(info, xValue, yValue, 1.0);

    p.NewPage();

    string[] lines = File.ReadAllLines(GetCurrentFileName());
}

```

```

const int FONTSIZE = 8;
const int MARGIN   = 36;
const int YSTART   = 750;

int yCurrent = YSTART;
bool showLine = false;

// Avoid false positive by building our
// markerBegin and markerEnd strings up dynamically

string tag           = "DemoThree";
string markerBegin = "// begin: " + tag;
string markerEnd   = "// end: "   + tag;

bool needSetFont = true;

for(int i = 0; i < lines.Length; i ++)
{
    if(!showLine)
    {
        if(lines[i].Contains(markerBegin))
            showLine = true;
    }
    else
    {
        if(lines[i].Contains(markerEnd))
            showLine = false;
    }

    if(showLine)
    {
        if(needSetFont)
        {
            p.SetFont(PDF.Font.COURIER, FONTSIZE);
            needSetFont = false;
        }

        p.ShowTextXY(lines[i], MARGIN, yCurrent);
        yCurrent -= FONTSIZE;

        if(yCurrent <= MARGIN)
        {
            p.NewPage();
            needSetFont = true;
            yCurrent = YSTART;
        }
    }
}
}
}
}

```