using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Intersect

{

class Program

{

static void Main(string[] args)

{

Point p = new Point(10, 30);

Point p1 = new Point(10, 45);

Point p2 = new Point(10, 60);

Point p3 = new Point(10, 85);

Point p4 = new Point(10, 100);

Point p5 = new Point(30, 100);

Point p6 = new Point(50, 100);

Point p7 = new Point(50, 70);

Point p8 = new Point(50, 50);

Point p9 = new Point(0, 50);

Line l01 = new Line(p, p1);

Line l12 = new Line(p1, p2);

Line l23 = new Line(p2, p3);

Line l34 = new Line(p3, p4);

Line l45 = new Line(p4, p5);

Line l56 = new Line(p5, p6);

Line l67 = new Line(p6, p7);

Line l78 = new Line(p7, p8);

Line l89 = new Line(p8, p9);

List<Line> output = new List<Line>();

output.Add(l01);

output.Add(l12);

output.Add(l23);

output.Add(l34);

output.Add(l45);

output.Add(l56);

output.Add(l67);

output.Add(l78);

output.Add(l89);

int count = 0;

foreach (var Line in output)

{

//Point intersection = l.GetIntersectionWith(

// Line.FromPoints(new Point(3, 0), new Point(0, 4)));

//Line line = Line.FromPoints(new Point(0, 0), new Point(3, 4));

// if()

Line.ToPrint();

count++;

}

l01.ToPrint();

Console.WriteLine(count);

Console.ReadKey();

}

}

public class Point

{

internal float x, y;

public Point()

{

x = y = 0;

}

public Point(float x, float y)

{

this.x = x;

this.y = y;

}

}

public class Line

{

internal Point a, b;

public Line(Point p, Point p1)

{

a = p;

b = p1;

}

internal void ToPrint()

{

Console.WriteLine("Line Begin From(x,y):({0},{1})", a.x, a.y);

Console.WriteLine("Line End In (x,y):({0},{1})", b.x, b.y);

}

}

}