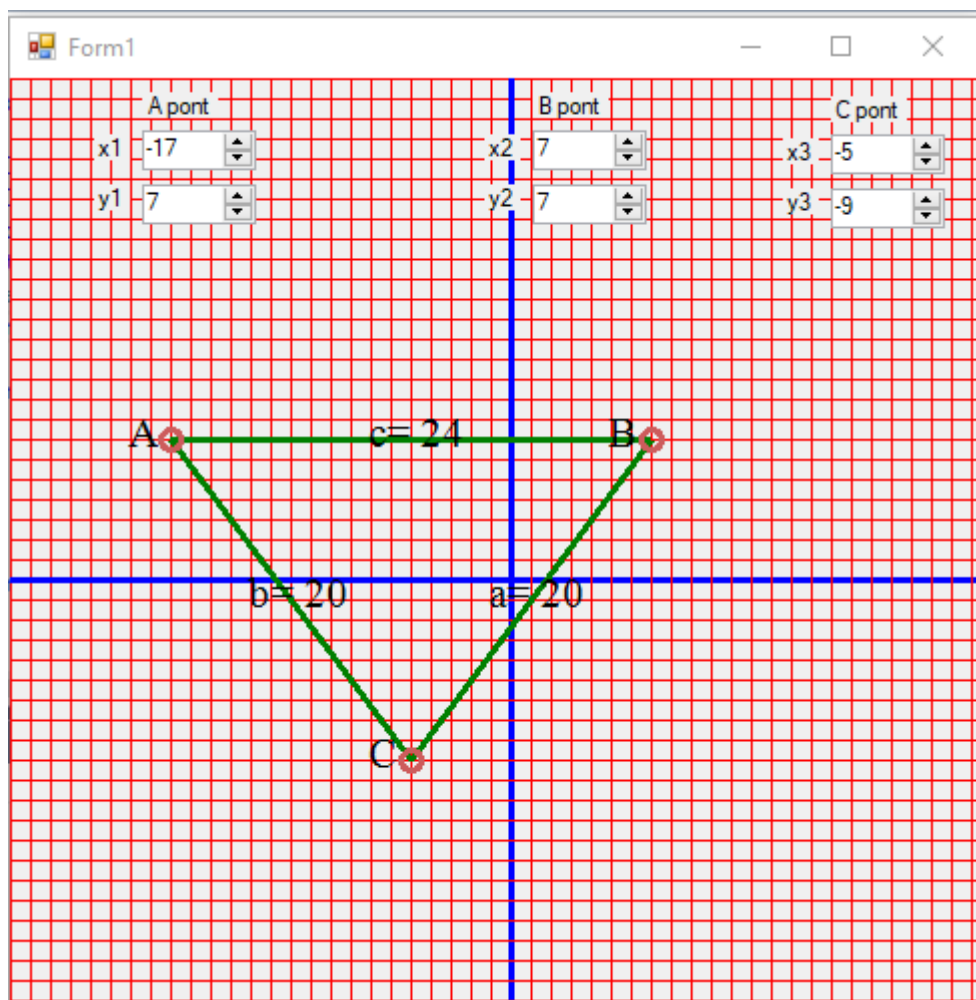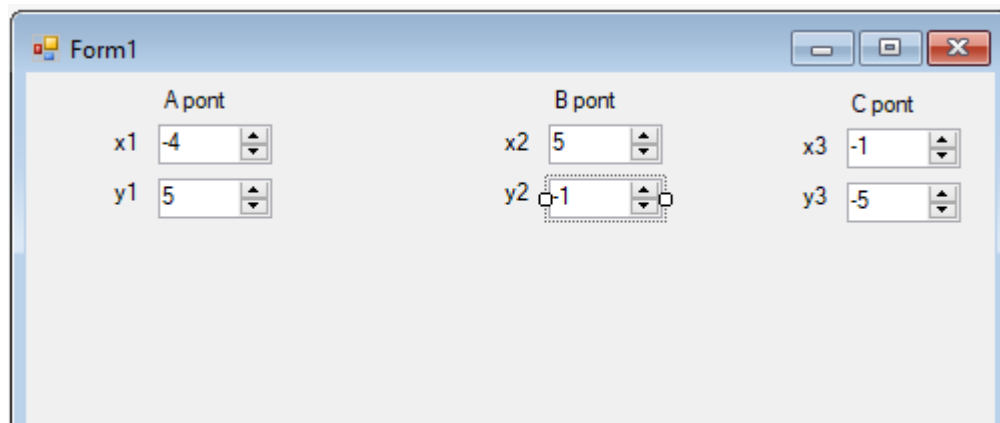Készítsünk programot, amely ábrázol egy háromszöget a koordinátarendszerben a csúcspontok koordinátáinak ismeretében. Lehessen változtatni a pontok helyzetét és írja ki az oldalak hosszát is!

Minta:

Megoldás: Az elemek tulajdonságait állítsuk be az alábbi táblázat szerint!

numericUpDown	Value	Maximum	Minimum	Name
numericUpDown1	-4	25	-25	a1
numericUpDown2	5	25	-25	a2
numericUpDown3	5	25	-25	b1
numericUpDown4	-1	25	-25	b2
numericUpDown5	-1	25	-25	c1
numericUpDown6	-5	25	-25	c2

Programkód:

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace szakasz
{
    public partial class Form1 : Form
    {
        float x = 500, y = 500;
        float x1 = -4, y1 = 5, x2 = 6, y2 = -1,x3=-1,y3=-5,f1,f2;
        double a, b, c;
        string oldal = "";
        Pen toll = new Pen(Color.Red, 1);
        int i;

        private void b2_ValueChanged(object sender, EventArgs e)
        {
            Invalidate();
            Update();
        }

        private void b1_ValueChanged(object sender, EventArgs e)
        {
            Invalidate();
            Update();
        }


        private void a1_ValueChanged(object sender, EventArgs e)
        {
            Invalidate();
            Update();
        }

        private void a2_ValueChanged(object sender, EventArgs e)
        {
            Invalidate();
            Update();
        }

        private void c1_ValueChanged(object sender, EventArgs e)
        {

            Invalidate();
            Update();
        }
private void c2_ValueChanged(object sender, EventArgs e)
        {
            Invalidate();
            Update();
        }


        public Form1()
        {
            InitializeComponent();
        }
```

```csharp
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = this.CreateGraphics();
    for (i = 0; i < x; i += 10)
    {
        if(i==250)
        {
            toll = new Pen(Color.Blue, 3);
        }
        else
        {
            toll = new Pen(Color.Red, 1);
        }
        g.DrawLine(toll, i, 0, i, y);
        g.DrawLine(toll, 0, i, x, i);

    }
    x1 = Convert.ToInt32(a1.Value);
    y1 = Convert.ToInt32(a2.Value);
    x2 = Convert.ToInt32(b1.Value);
    y2 = Convert.ToInt32(b2.Value);
    x3 = Convert.ToInt32(c1.Value);
    y3 = Convert.ToInt32(c2.Value);
    toll = new Pen(Color.Green, 3);
    g.DrawLine(toll, 250+x1*10, 250-y1*10, 250 + x2 * 10, 250 - y2 * 10);
    g.DrawLine(toll, 250 + x1 * 10, 250 - y1 * 10, 250 + x3 * 10, 250 - y3 * 10);
    g.DrawLine(toll, 250 + x2 * 10, 250 - y2 * 10, 250 + x3 * 10, 250 - y3 * 10);
    toll = new Pen(Color.IndianRed,3);
    g.DrawEllipse(toll, 250 + x1 * 10 -5, 250 - y1 * 10-5 , 10, 10);
    g.DrawEllipse(toll, 250 + x2 * 10 - 5, 250 - y2 * 10 - 5, 10, 10);
    g.DrawEllipse(toll, 250 + x3 * 10 - 5, 250 - y3 * 10 - 5, 10, 10);
    SolidBrush ecset = new SolidBrush(Color.Black);
    oldal = "";
    f1 = (x1 + x2) / 2;f2 = (y1 + y2) / 2;
    c = Math.Sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
    c = Math.Round(c, 2);
    oldal += "c= " + Convert.ToString(c);
g.DrawString(oldal, new Font("Times New Roman", 16), ecset, 250 + f1 * 10 - 25, 250 - f2 * 10 - 15);

    oldal = "";
    f1 = (x1 + x3) / 2; f2 = (y1 + y3) / 2;
    b = Math.Sqrt((x3 - x1) * (x3 - x1) + (y3 - y1) * (y3 - y1));
    b = Math.Round(b, 2);
    oldal += "b= " + Convert.ToString(b);
g.DrawString(oldal, new Font("Times New Roman", 16), ecset, 250 + f1 * 10 - 25, 250 - f2 * 10 - 15);

     oldal = "";
    f1 = (x2 + x3) / 2; f2 = (y2 + y3) / 2;
    a = Math.Sqrt((x3 - x2) * (x3 - x2) + (y3 - y2) * (y3 - y2));
    a = Math.Round(a, 2);
    oldal += "a= " + Convert.ToString(a);
g.DrawString(oldal, new Font("Times New Roman", 16), ecset, 250 + f1 * 10 - 25, 250 - f2 * 10 - 15);


g.DrawString("A",new Font ("Times New Roman", 16),ecset, 250 + x1 * 10 - 25, 250 - y1 * 10 - 15);
g.DrawString("B", new Font("Times New Roman", 16), ecset, 250 + x2 * 10 - 25, 250 - y2 * 10 - 15);
g.DrawString("C", new Font("Times New Roman", 16), ecset, 250 + x3 * 10 - 25, 250 - y3 * 10 - 15);

    }
  }
}
```