Лабораторная работа №10

Выполнил: Новиков Даниил

Группа: ФИб-2301

**Задачи для аудиторного решения.**

*Задача 1.* Даны 3 точки A, B и C. Определить, лежат ли они на одной прямой.

Код программы:

class Program

{

static void Main(string[] args)

{

double ax, ay, bx, by, cx, cy;//координаты точек

Console.Write("Введите ax:");

ax = double.Parse(Console.ReadLine());

Console.Write("Введите ay:");

ay = double.Parse(Console.ReadLine());

Console.Write("Введите bx:");

bx = double.Parse(Console.ReadLine());

Console.Write("Введите by:");

by = double.Parse(Console.ReadLine());

Console.Write("Введите cx:");

cx = double.Parse(Console.ReadLine());

Console.Write("Введите cy:");

cy = double.Parse(Console.ReadLine());

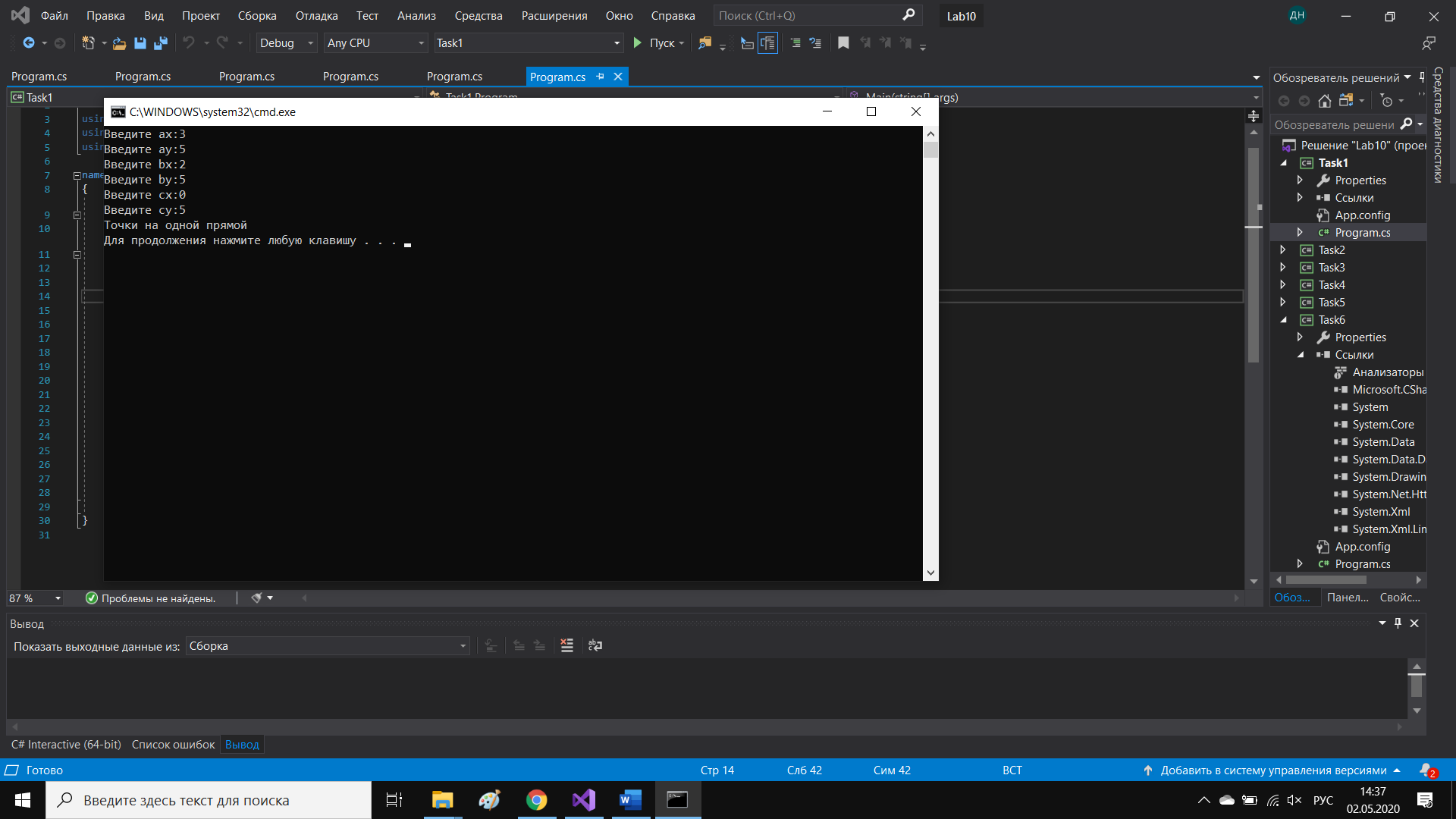
if ((cx - ax) \* (by - ay) == (cy - ay)\*(bx-ax)) Console.WriteLine("Точки на одной прямой");

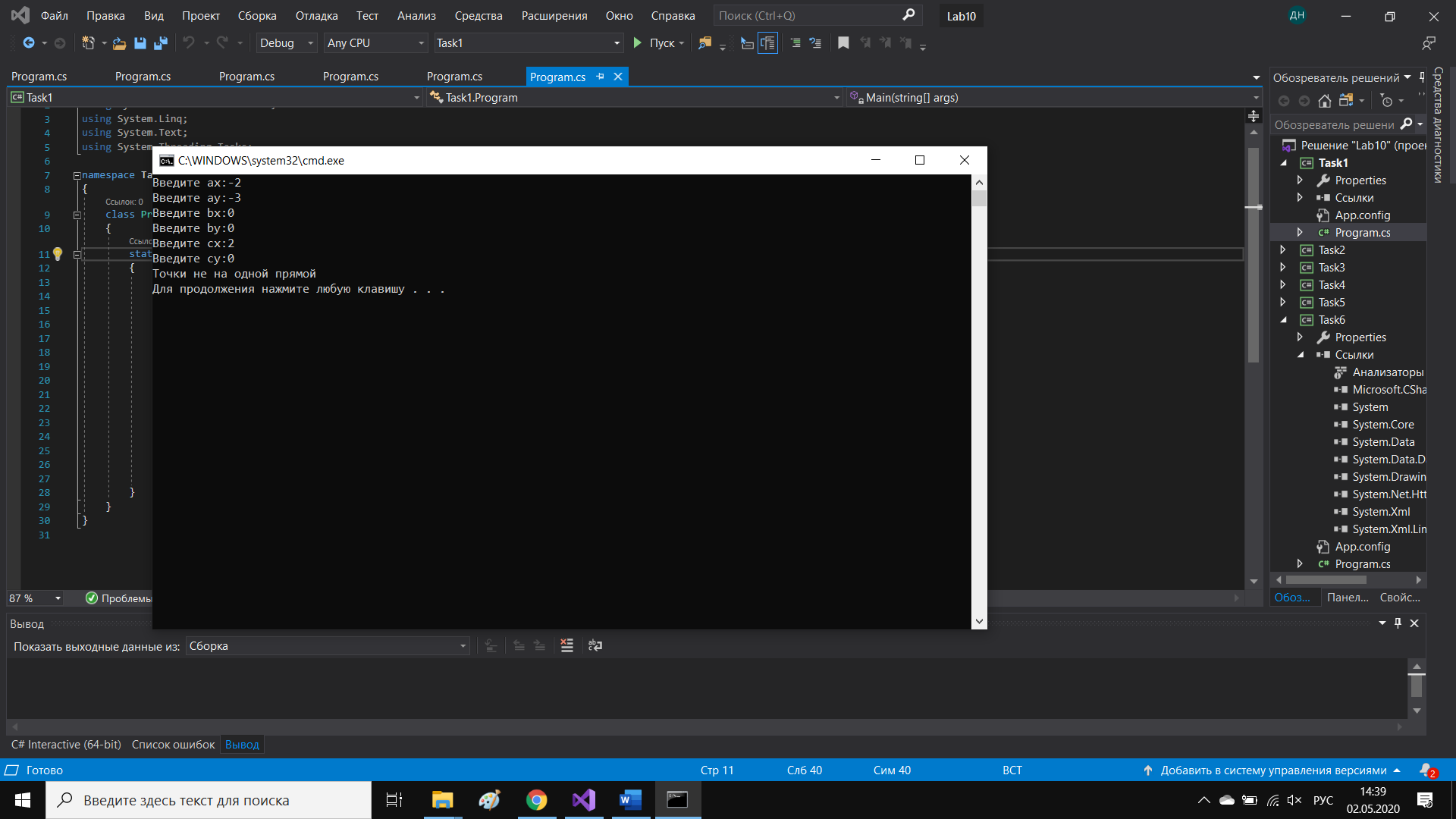
else Console.WriteLine("Точки не на одной прямой");

}

}

Результат работы программы:





*Задача 2.* Даны 3 точки A, B и C, не лежащие на одной прямой. Определить, является обход A→B→C обходом по часовой стрелке или против часовой стрелки.

Код программы:

class Program

{

static void Main(string[] args)

{

double ax, ay, bx, by, cx, cy;//координаты точек

Console.Write("Введите ax:");

ax = double.Parse(Console.ReadLine());

Console.Write("Введите ay:");

ay = double.Parse(Console.ReadLine());

Console.Write("Введите bx:");

bx = double.Parse(Console.ReadLine());

Console.Write("Введите by:");

by = double.Parse(Console.ReadLine());

Console.Write("Введите cx:");

cx = double.Parse(Console.ReadLine());

Console.Write("Введите cy:");

cy = double.Parse(Console.ReadLine());

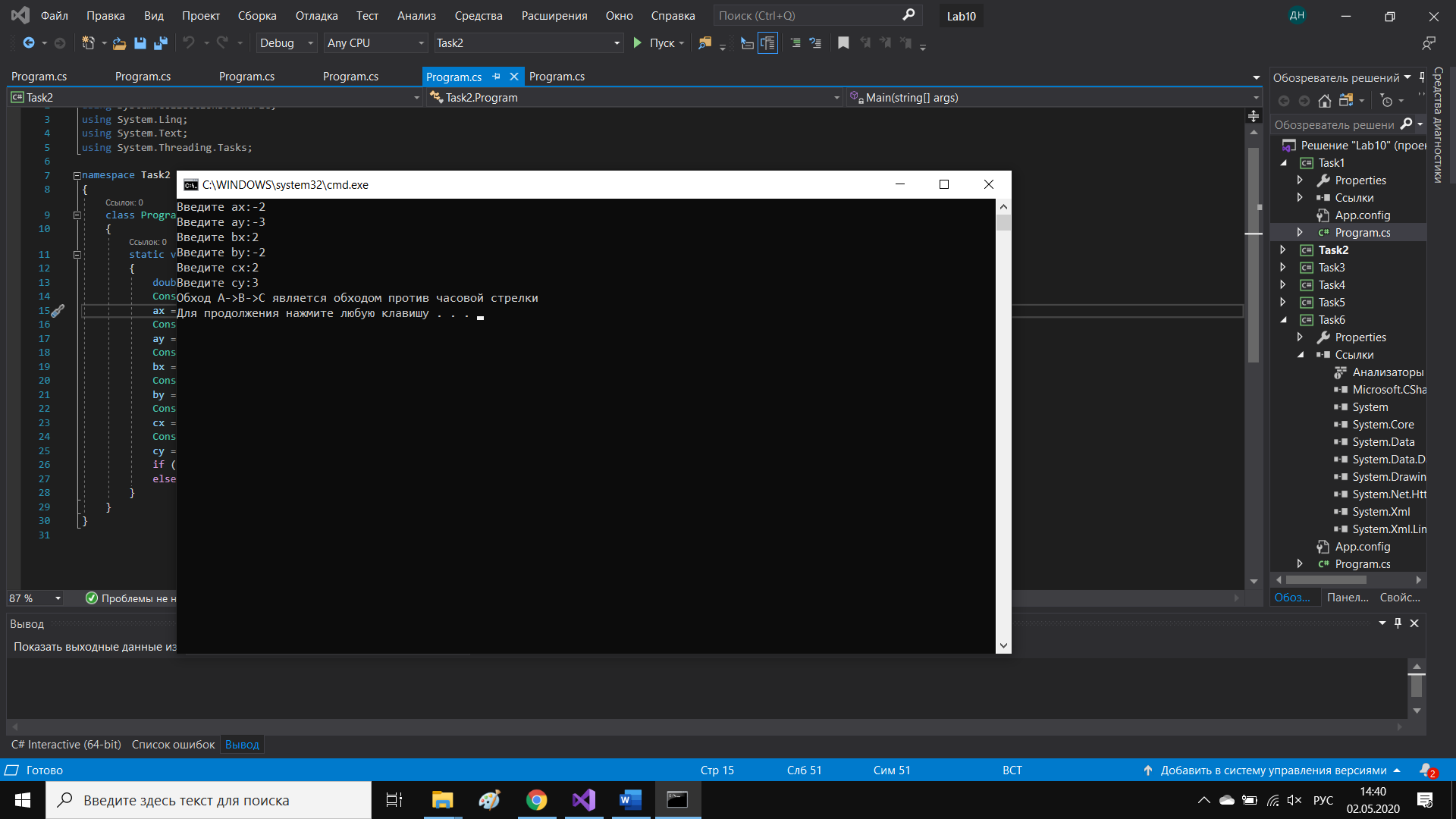
if ((bx - ax) \* (cy - ay) - (by - ay) \* (cx - ax)<0) Console.WriteLine("Обход A->B->C является обходом по часовой стрелке");

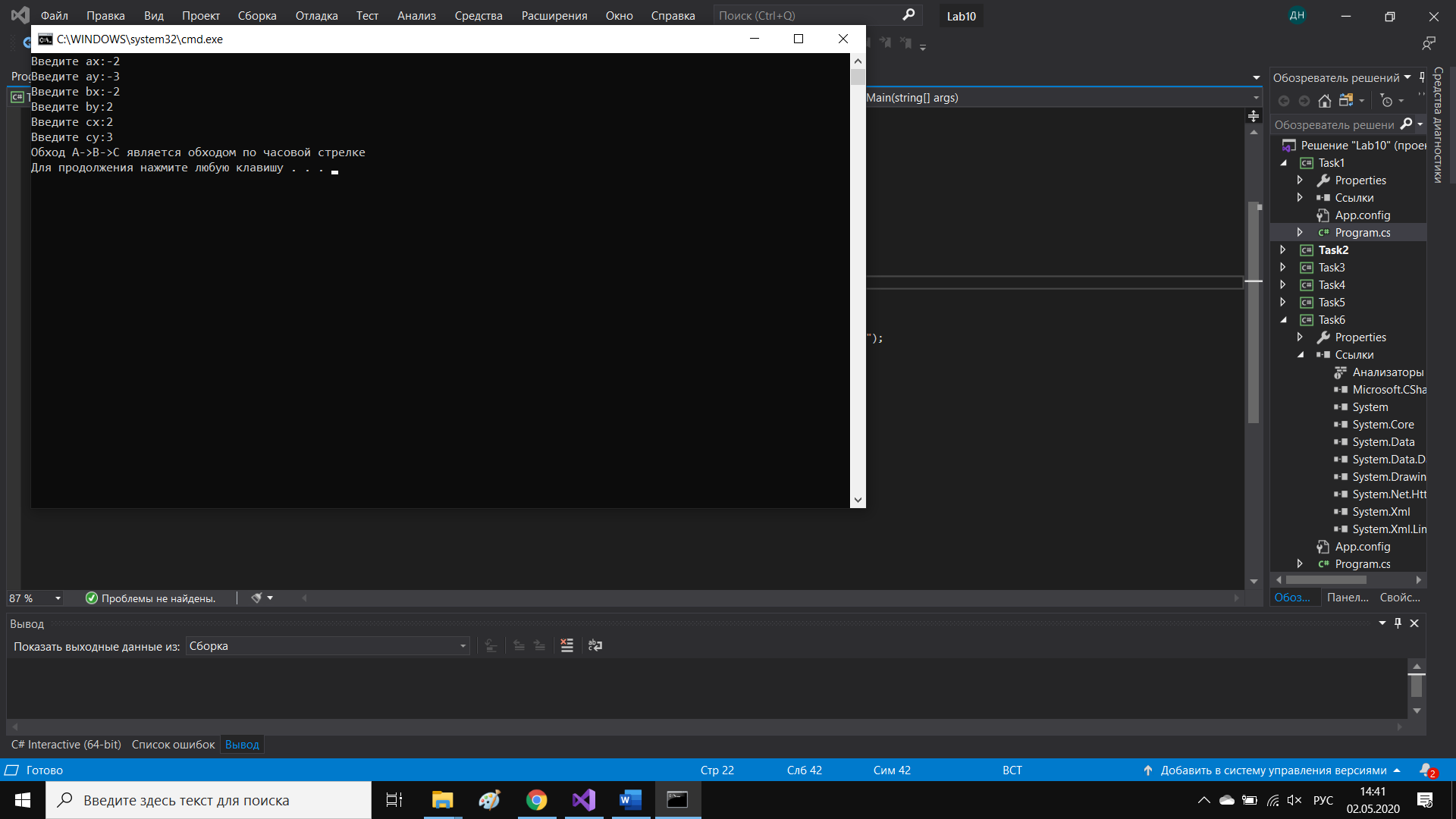
else Console.WriteLine("Обход A->B->C является обходом против часовой стрелки");

}

}

Результат работы программы:





*Задача 3.* Даны 4 точки A, B, C и D. Определить, пересекаются ли отрезки AC и BD.

Код программы:

class Program

{

static void Main(string[] args)

{

PointF A, B, C, D;//координаты точек

A = new PointF();

B = new PointF();

C = new PointF();

D = new PointF();

Console.Write("Введите ax:");

A.X = float.Parse(Console.ReadLine());

Console.Write("Введите ay:");

A.Y = float.Parse(Console.ReadLine());

Console.Write("Введите bx:");

B.X = float.Parse(Console.ReadLine());

Console.Write("Введите by:");

B.Y = float.Parse(Console.ReadLine());

Console.Write("Введите cx:");

C.X = float.Parse(Console.ReadLine());

Console.Write("Введите cy:");

C.Y = float.Parse(Console.ReadLine());

Console.Write("Введите dx:");

D.X = float.Parse(Console.ReadLine());

Console.Write("Введите dy:");

D.Y = float.Parse(Console.ReadLine());

if (IsLineCross(A,C,B,D))

Console.WriteLine("Отрезки AC и BD пересекаются");

else Console.WriteLine("Отрезки AC и BD не пересекаются");

}

static bool IsLineCross(PointF A, PointF B, PointF C, PointF D)

{

float vec1 = VectorMult(D.X - C.X, D.Y - C.Y, A.X - C.X, A.Y - C.Y);

float vec2 = VectorMult(D.X - C.X, D.Y - C.Y, B.X - C.X, B.Y - C.Y);

float vec3 = VectorMult(B.X - A.X, B.Y - A.Y, C.X - A.X, C.Y - A.Y);

float vec4 = VectorMult(B.X - A.X, B.Y - A.Y, D.X - A.X, D.Y - A.Y);

if (vec1 \* vec2 < 0 && vec3 \* vec4 < 0)

return true;

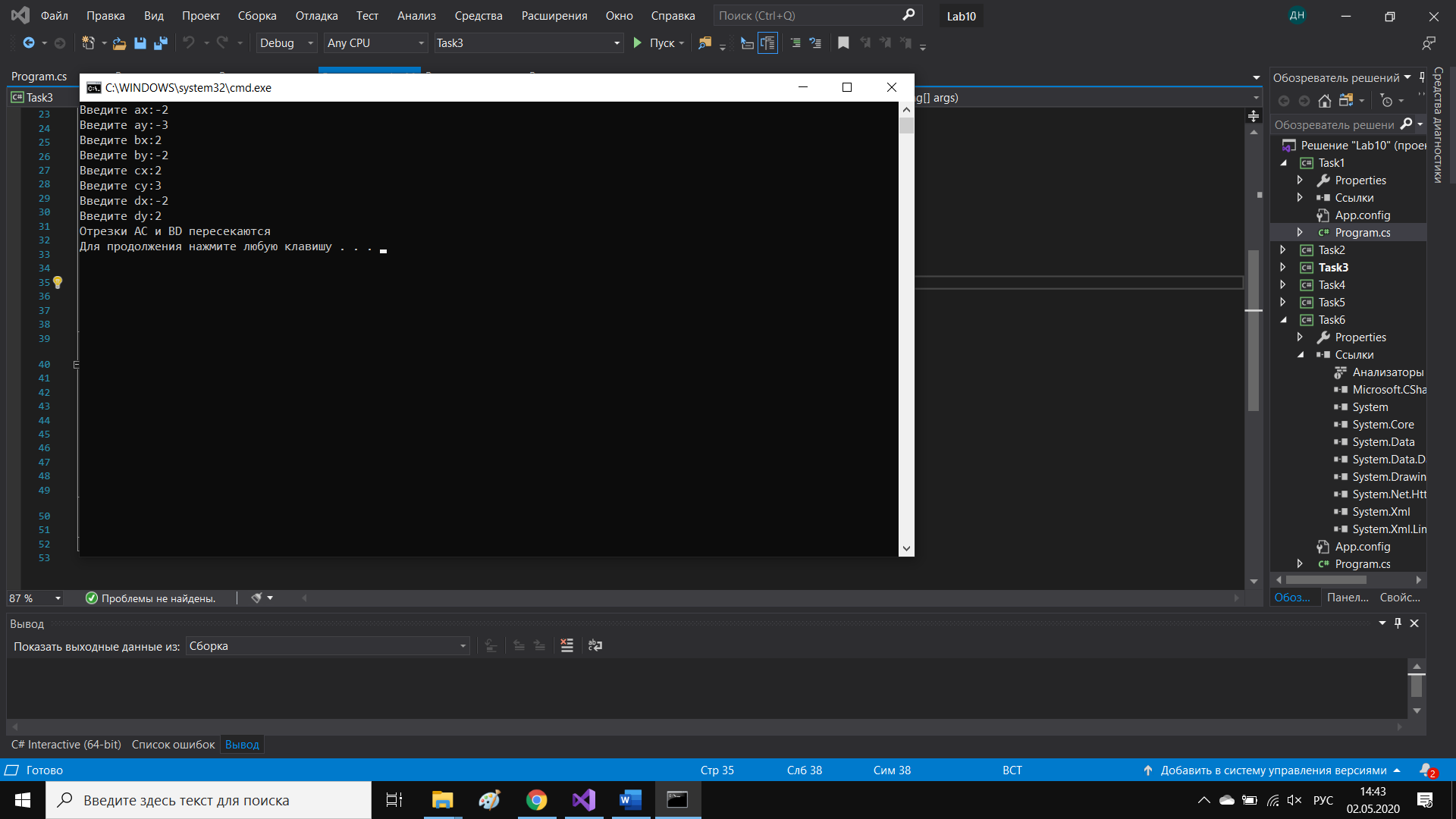
else return false;

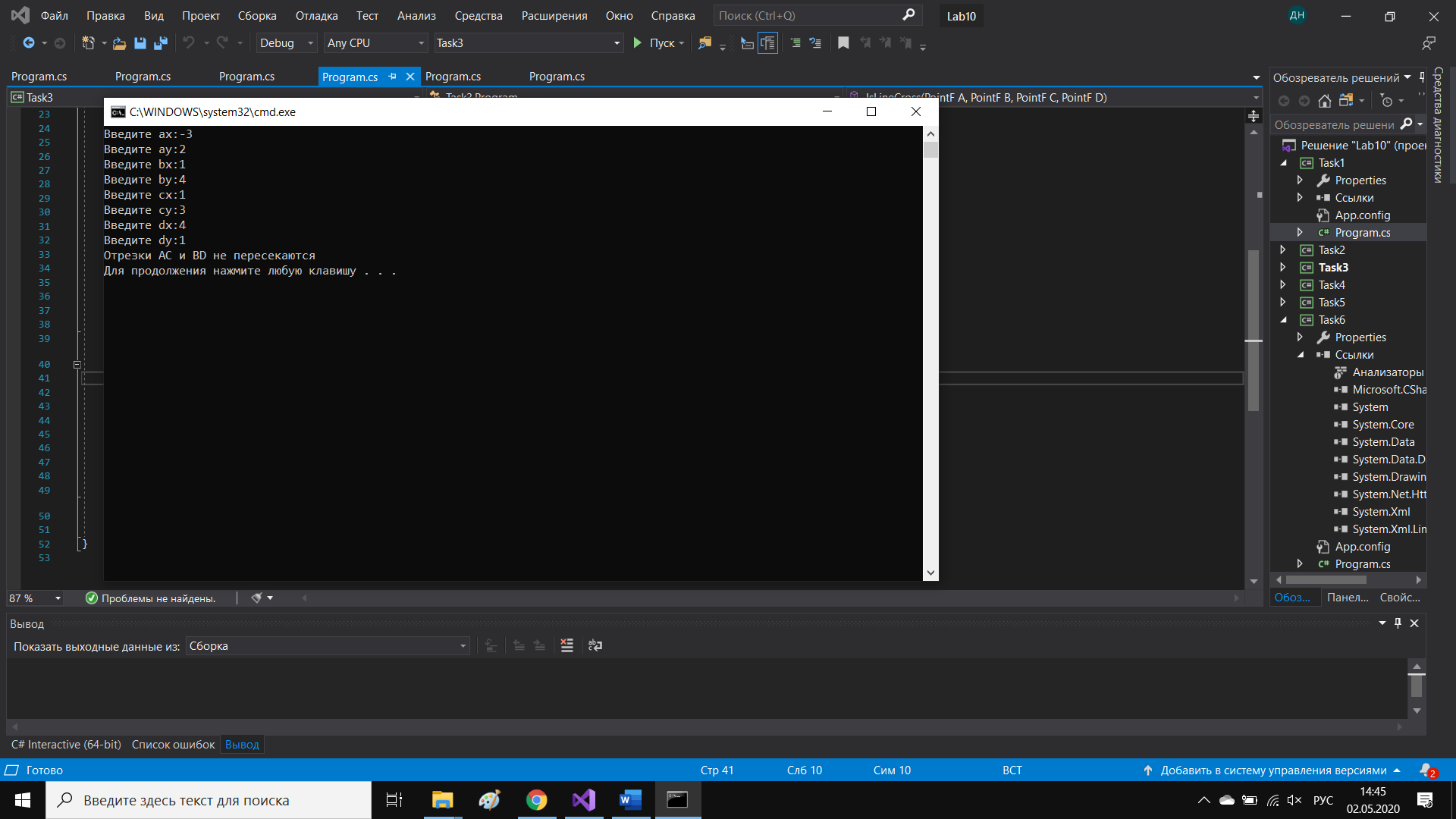
}

static float VectorMult(float x1, float y1, float x2, float y2) => x1 \* y2 - x2 \* y1;

}

Результат работы программы:





*Задача 4.* Даны 4 точки A, B, C и D. Определить, является ли четырехугольник ABCD выпуклым.

Код программы:

class Program

{

static void Main(string[] args)

{

PointF A, B, C, D;//координаты точек

A = new PointF();

B = new PointF();

C = new PointF();

D = new PointF();

Console.Write("Введите ax:");

A.X = float.Parse(Console.ReadLine());

Console.Write("Введите ay:");

A.Y = float.Parse(Console.ReadLine());

Console.Write("Введите bx:");

B.X = float.Parse(Console.ReadLine());

Console.Write("Введите by:");

B.Y = float.Parse(Console.ReadLine());

Console.Write("Введите cx:");

C.X = float.Parse(Console.ReadLine());

Console.Write("Введите cy:");

C.Y = float.Parse(Console.ReadLine());

Console.Write("Введите dx:");

D.X = float.Parse(Console.ReadLine());

Console.Write("Введите dy:");

D.Y = float.Parse(Console.ReadLine());

if (CheckConvexPolygon(A, B, C, D)) Console.WriteLine("Четырехугольник ABCD является выпуклым");

else Console.WriteLine("Четырехугольник ABCD не является выпуклым");

}

/// <summary>

/// проверка на то, что 3 точки на одной прямой

/// </summary>

/// <param name="A"></param>

/// <param name="B"></param>

/// <param name="C"></param>

/// <returns></returns>

static bool OnSameLine(PointF A,PointF B,PointF C)

{

if ((C.X - A.X) \* (B.Y - A.Y) == (C.Y - A.Y) \* (B.X - A.X))

return true;

else return false;

}

static int IsClockWise(PointF A, PointF B, PointF C)

{

if ((B.X - A.X) \* (C.Y - A.Y) - (B.Y - A.Y) \* (C.X - A.X) < 0) return 1;//по часовой стрелке

else return -1;//против часовой стрелки

}

static bool CheckConvexPolygon(PointF A, PointF B, PointF C,PointF D)

{

if (OnSameLine(A, B, C) || OnSameLine(B, C, D) || OnSameLine(A, C, D) //если какие-то 3 точки лежат на одной прямой или все 4 лежат на одной прямой

|| (OnSameLine(A, B, C) && OnSameLine(B, C, D)))

return true;

else//проверяем каждую тройку на обход по часовой или против часовой стрелки

{

if ((IsClockWise(A, B, C) == 1 && IsClockWise(B, C, D) == 1 && IsClockWise(C, D, A) == 1 && IsClockWise(D, A, C) == 1) ||

(IsClockWise(A, B, C) == -1 && IsClockWise(B, C, D) == -1 && IsClockWise(C, D, A) == -1 && IsClockWise(D, A, C) == -1))

return true;//если каждой тройки один обход

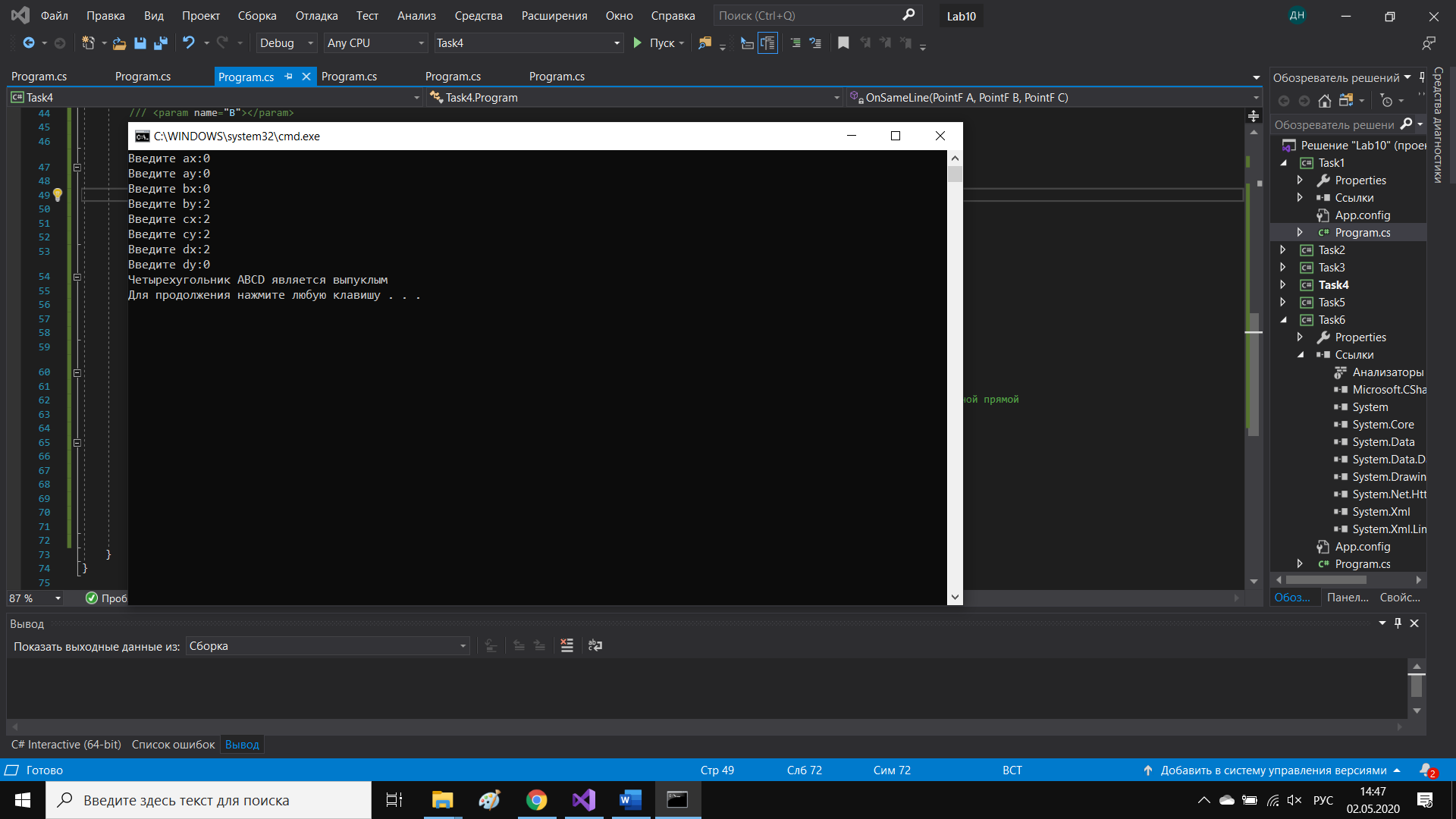
else return false;

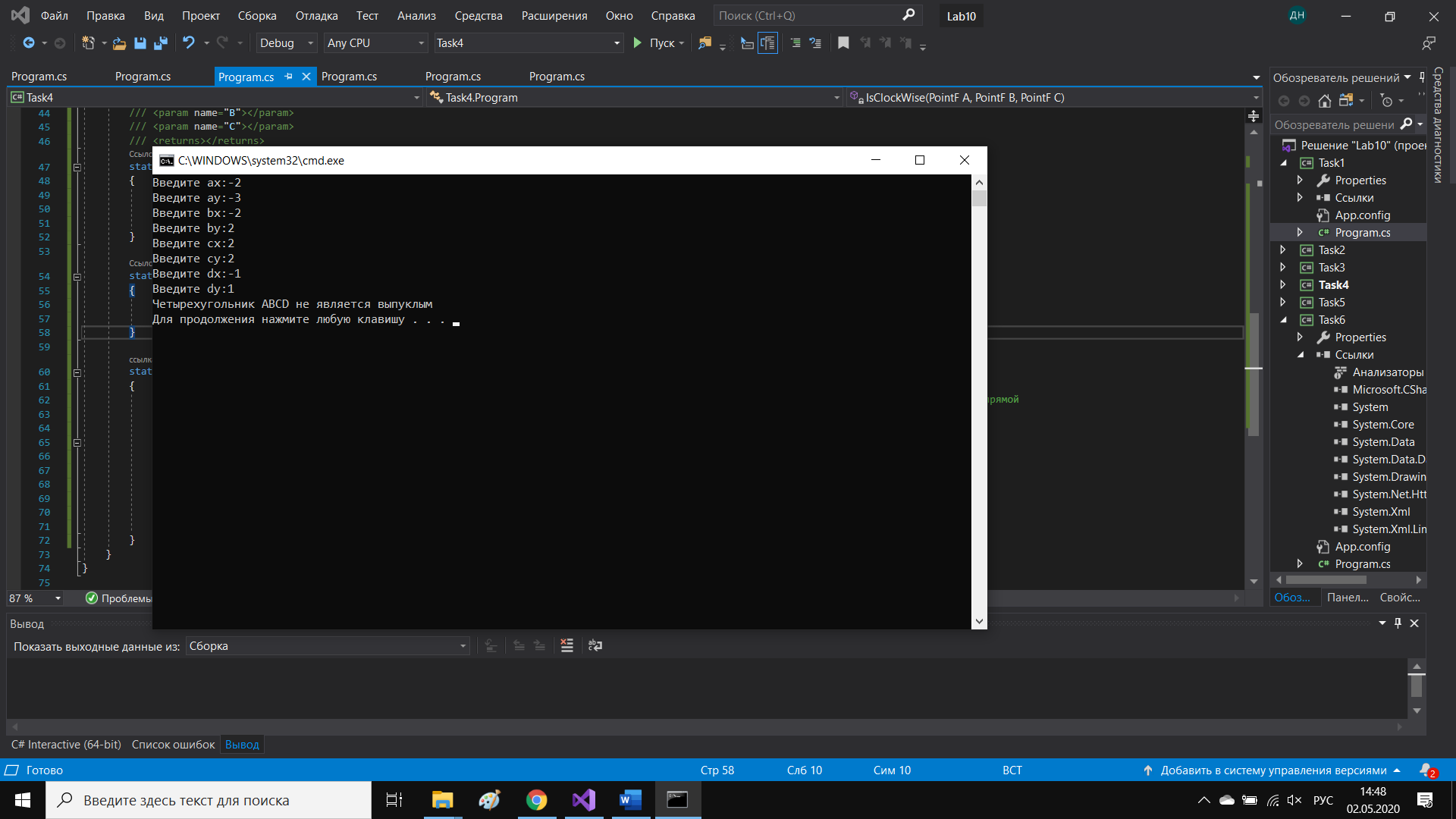
}

}

}

Результат работы программы:





**Задачи для самостоятельного решения.**

*Задача 1.* Три точки A, B и C, лежат на одной прямой. Определите, какая из

них лежит между двумя другими.

Код программы:

class Program

{

static void Main(string[] args)

{

PointF A = new PointF();

PointF B = new PointF();

PointF C = new PointF();

int n;//количество проверок

string tmp;//строка с координатами

char[] answers;

using(StreamReader sr=new StreamReader("input.txt"))

{

n = int.Parse(sr.ReadLine());

answers = new char[n];

for(int i=0;i<n;i++)//считываем кажду строку

{

tmp = sr.ReadLine();

string[] coordinates = Regex.Split(tmp, " ");

A.X = float.Parse(coordinates[0]);

A.Y = float.Parse(coordinates[1]);

B.X = float.Parse(coordinates[2]);

B.Y = float.Parse(coordinates[3]);

C.X = float.Parse(coordinates[4]);

C.Y = float.Parse(coordinates[5]);

answers[i] = GetMidPoint(A, B, C);

}

}

using (StreamWriter sw = new StreamWriter("output.txt"))

{

for(int i=0;i<n;i++)

{

sw.WriteLine(answers[i]);

}

}

}

static char GetMidPoint(PointF A,PointF B,PointF C)

{

float ab, bc, ac;//расстояния между точками

ab = (float)Math.Sqrt(Math.Pow(B.X - A.X, 2) + Math.Pow(B.Y - A.Y, 2));

bc = (float)Math.Sqrt(Math.Pow(C.X - B.X, 2) + Math.Pow(C.Y - B.Y, 2));

ac = (float)Math.Sqrt(Math.Pow(C.X - A.X, 2) + Math.Pow(C.Y - A.Y, 2));

if (ab == bc + ac) return 'C';

else if (ac == ab + bc) return 'B';

else if (bc == ab + ac) return 'A';

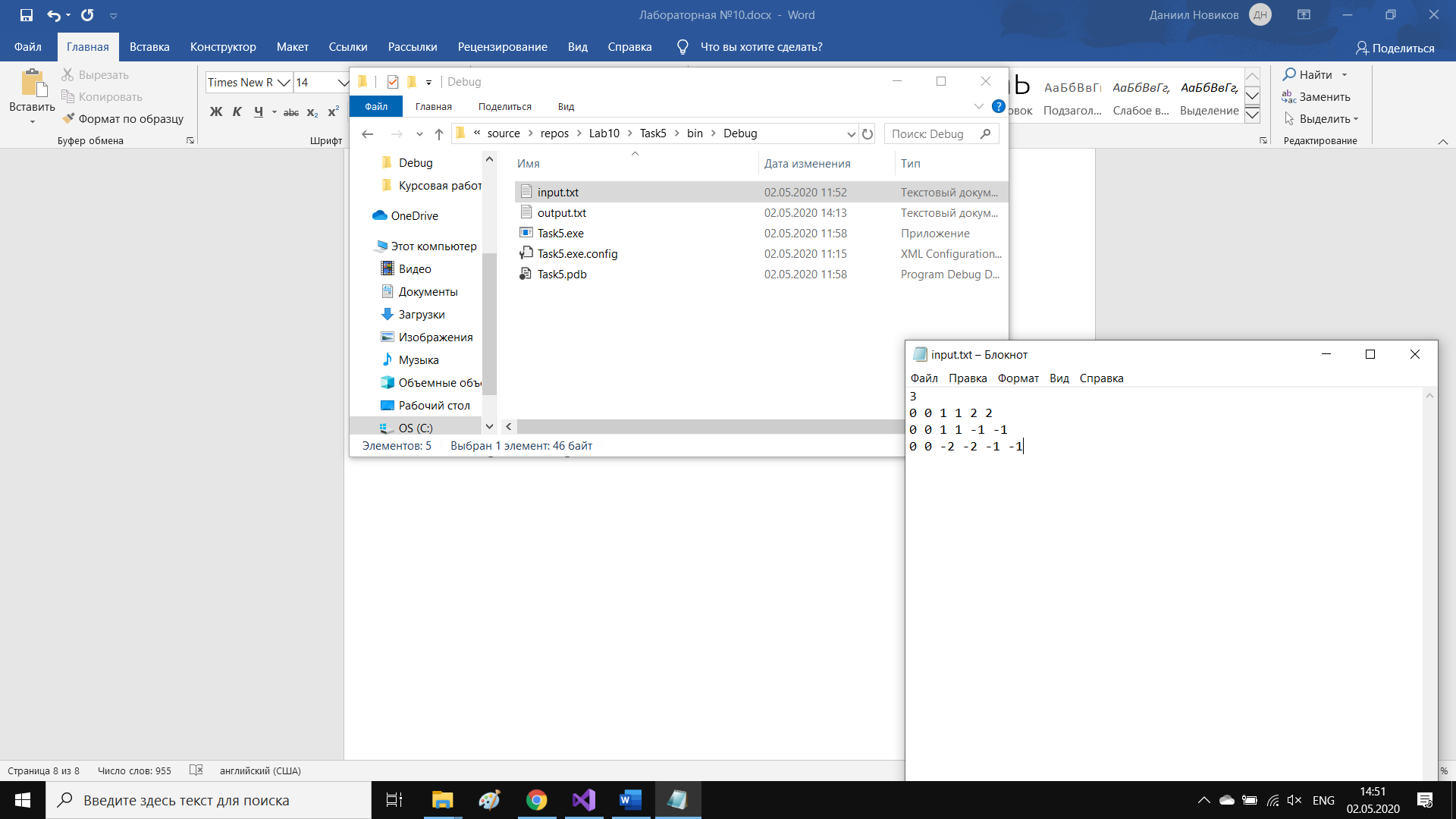
else throw new ArgumentException("Точки не лежат на одной прямой");

}

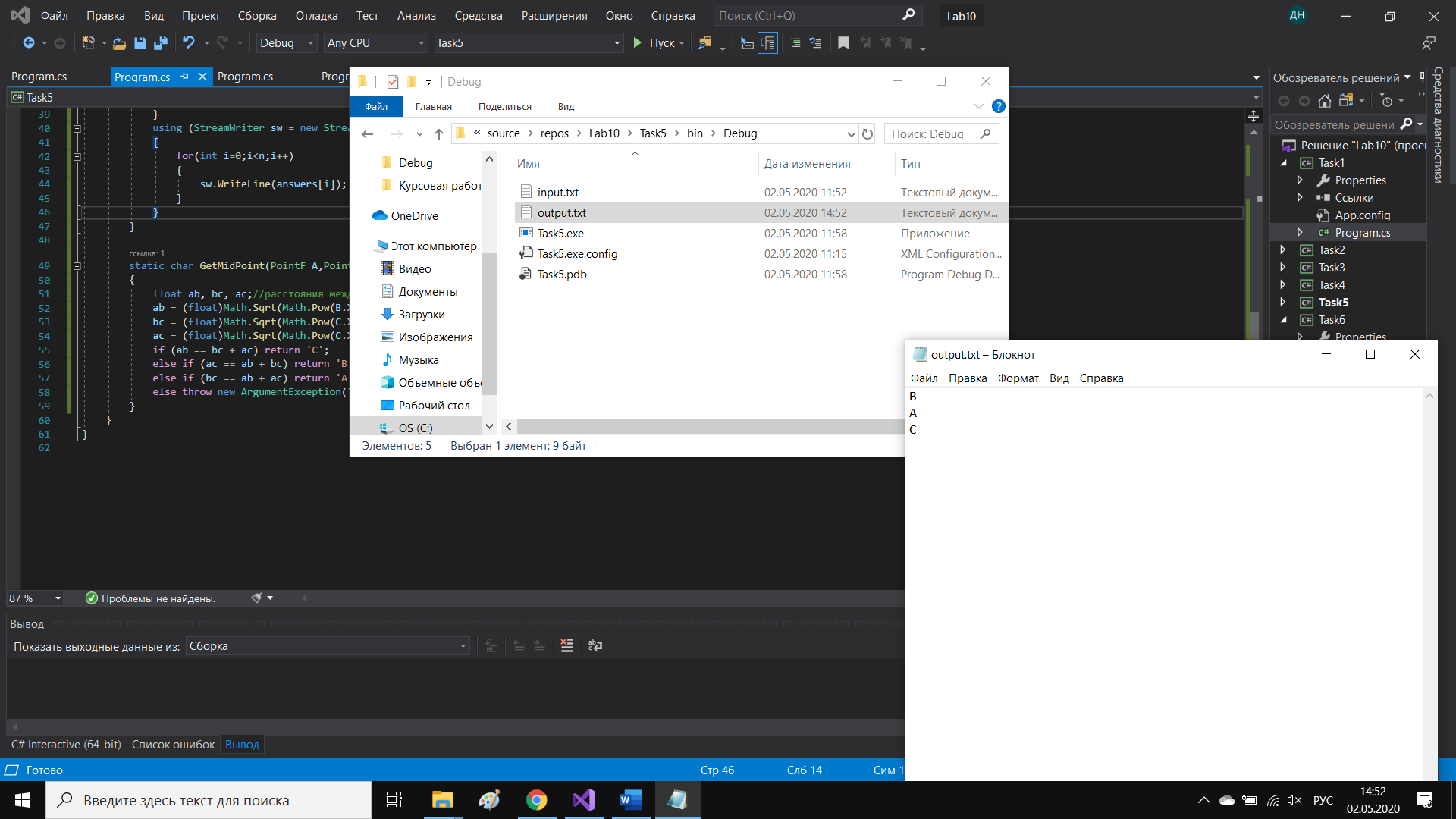
}

Результат работы программы:

Содержимое input.txt:



Содержимое output.txt после выполнения программы:



*Задача 2.* Даны пересекающиеся 2 прямые на плоскости, найти точку их

пересечения.

Код программы:

class Program

{

static void Main(string[] args)

{

PointF A = new PointF();

PointF B = new PointF();

PointF C = new PointF();

PointF D = new PointF();

string tmp;//строка с координатами

using (StreamReader sr = new StreamReader("input.txt"))

{

tmp = sr.ReadLine();

tmp = Regex.Replace(tmp, @"\.", ",");

string[] coordinates = Regex.Split(tmp, " ");

A.X = float.Parse(coordinates[0]);

A.Y = float.Parse(coordinates[1]);

B.X = float.Parse(coordinates[2]);

B.Y = float.Parse(coordinates[3]);

tmp = sr.ReadLine();

tmp = Regex.Replace(tmp, @"\.", ",");

coordinates = Regex.Split(tmp, " ");

C.X = float.Parse(coordinates[0]);

C.Y = float.Parse(coordinates[1]);

D.X = float.Parse(coordinates[2]);

D.Y = float.Parse(coordinates[3]);

}

PointF intersection = FindIntersectPoint(A, B, C, D);

string answer = String.Format("{0:0.0000}", intersection.X) +" "+ String.Format("{0:0.0000}", intersection.Y);

answer = Regex.Replace(answer, ",", ".");

using (StreamWriter sw = new StreamWriter("output.txt"))

{

sw.Write(answer);

}

}

static PointF FindIntersectPoint(PointF A, PointF B, PointF C, PointF D)

{

float dxAB = B.X - A.X;

float dyAB = B.Y - A.Y;

float dxCD = D.X - C.X;

float dyCD = D.Y - C.Y;

float den = (dyAB \* dxCD - dxAB \* dyCD);

float t1 = ((A.X - C.X) \* dyCD + (C.Y - A.Y) \* dxCD) / den;//находим параметры для прямой

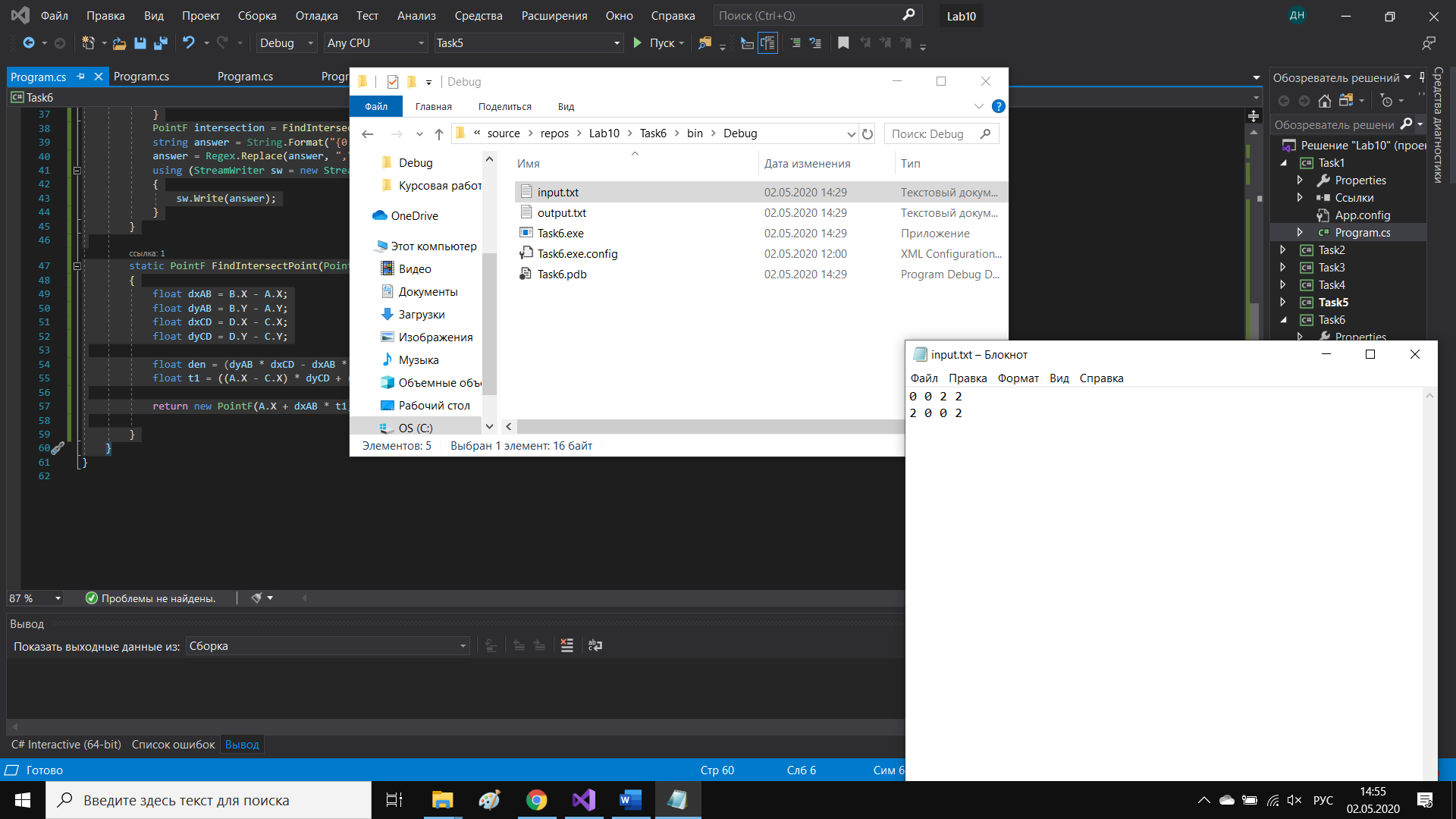
return new PointF(A.X + dxAB \* t1, A.Y + dyAB \* t1);

}

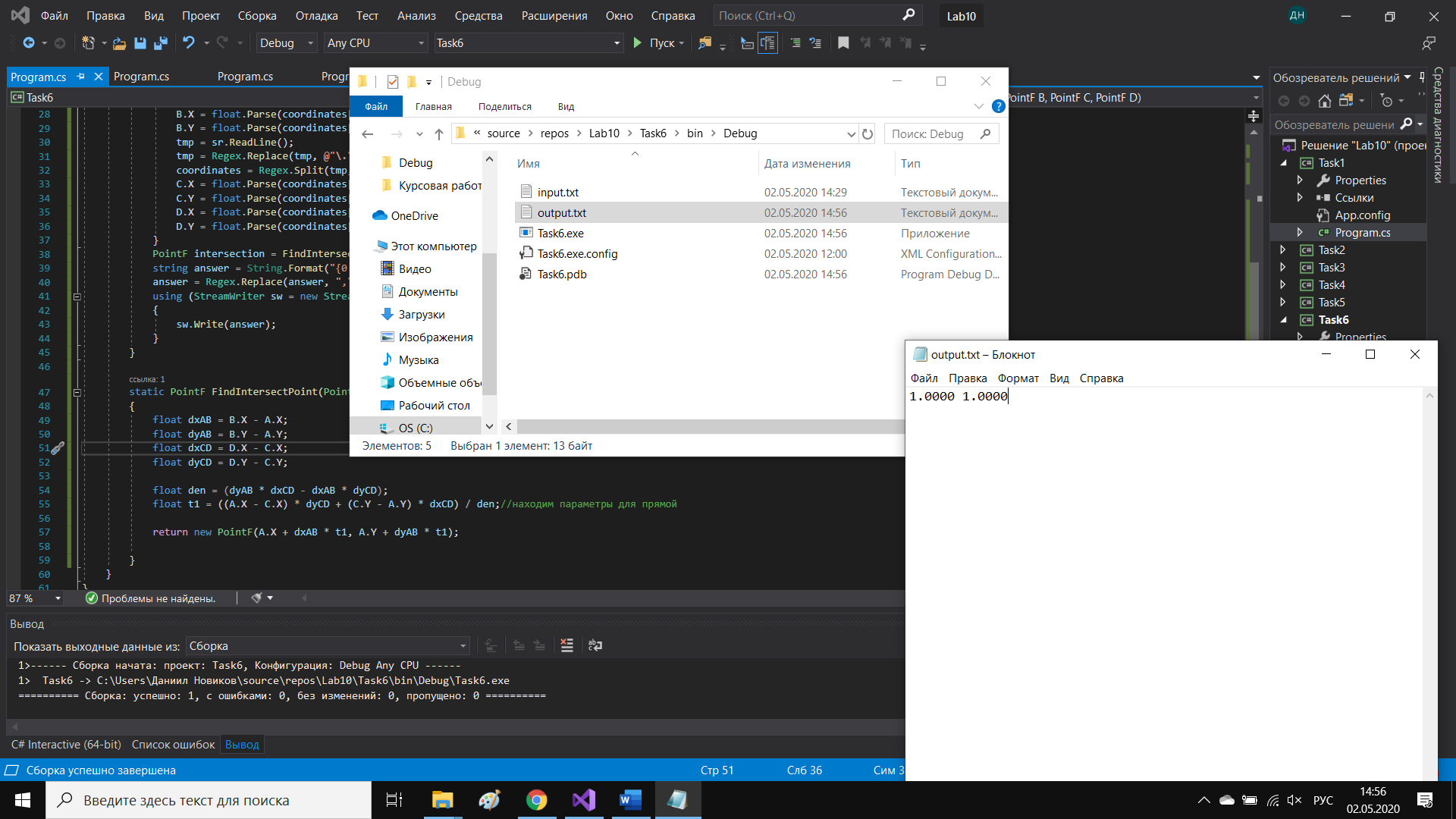
}

Результат работы программы:

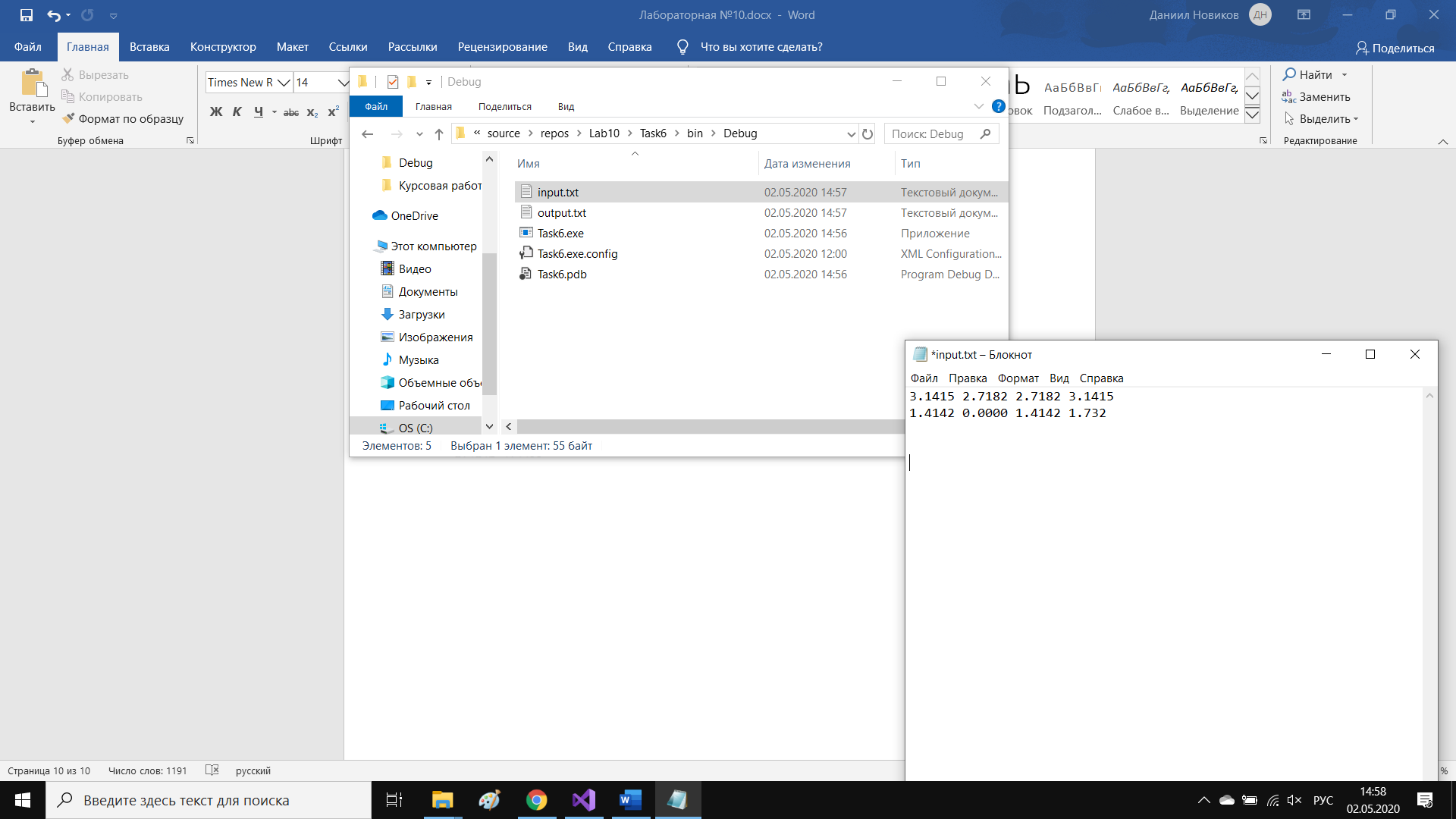
Содержимое input.txt:



Содержимое output.txt после выполнения программы:



Содержимое input.txt:



Содержимое output.txt после выполнения программы:

