

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

КУРСОВАЯ РАБОТА
по дисциплине «Программирование»
ТЕМА: ХРАНЕНИЕ И ОБРАБОТКА ЧИСЛОВЫХ ДАННЫХ НА ОСНОВЕ ФАЙЛОВ И
МАССИВОВ

Студент гр. 4354

Чучалин И. В.

Преподаватель

Калмычков В.А.

Санкт-Петербург

2024

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Чучалин И.В.

Группа 4354

Тема работы: хранение и обработка числовых данных на основе файлов и массивов.

Исходные данные: реализовать алгоритм на основе разбиения программы на набор функций с параметрами. Программа не должна содержать стандартных и библиотечных типов и связанных с ними функций. Должны быть учтены все составляющие формулировки задания и выполнены указанные действия, причем в качестве результата должны быть предложены все варианты, удовлетворяющие заданию. Программа должна предоставлять возможность неоднократного выполнения действий.

Содержание пояснительной записки:

Исходная формулировка задания, математическая постановка задачи, описание алгоритма, текст программы, вывод программы, выводы.

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 22.11.2024

Дата сдачи реферата:

Дата защиты реферата:

Студент

Чучалин И.В.

Преподаватель

Калмычков В.А.

АННОТАЦИЯ

В данной курсовой работе рассмотрена работа с числовыми данными, файлами, массивами и функциями на примере языка C++. Необходимо корректно составить математическую постановку задачи, обработать файл с исходными данными, при помощи двумерных или трёхмерных массивов организовать хранение данных и с помощью функций реализовать программу для решения геометрической задачи. Результатом выполнения курсовой работы служит программа, способная решать геометрическую задачу с разными начальными условиями и реагирующая на некорректные данные.

SUMMARY

In this course work, work with numeric data, files, arrays and functions is considered using the example of the C++ language. It is necessary to correctly formulate a mathematical statement of the problem, process the file with the initial data, organize data storage using two-dimensional or three-dimensional arrays and use functions to implement a program to solve a geometric problem. The result of the course work is a program capable of solving a geometric problem with different initial conditions and responding to incorrect data.

Содержание:	
Исходная формулировка задания.....	5
Ограничения на исходные данные.....	5
Математическая постановка задачи	5
Дано.....	5
Найти	5
Способ решения.....	5
Вычислим длины сторон:	5
Контрольный пример	6
Организация интерфейса пользователя.....	6
Макет ввода/вывода	6
Реализация вывода в консоль	7
Внутренний формат хранения данных	7
Работа с файлами.....	7
Описание функций	8
Синтаксис	8
Назначение	8
Описание алгоритма.....	9
Текст программы	9
Вывод программы.....	13
Выводы	15

Исходная формулировка задания

Дано N точек на плоскости. Для всех треугольников, образуемых любыми тремя точками, определить, является ли треугольник остроугольным.

Ограничения на исходные данные

Будем обрабатывать не больше 50 точек ($N \leq 50$).

Математическая постановка задачи

Дано N точек на плоскости.

Найти все остроугольные треугольники, образуемые любыми тремя точками.

Способ решения

Вычислим длины сторон:

$$|AB| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$|AC| = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}$$

$$|BC| = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$$

Вычислим внутренние углы по теореме косинусов:

$$\cos \angle A = \frac{|AB|^2 + |AC|^2 - |BC|^2}{2 \cdot |AB| \cdot |AC|}$$

$$\angle A = \arccos$$

$$\cos \angle B = \frac{|AB|^2 + |BC|^2 - |AC|^2}{2 \cdot |AB| \cdot |BC|}$$

$$\angle B = \arccos$$

$$\cos \angle C = \frac{|AC|^2 + |BC|^2 - |AB|^2}{2 \cdot |AC| \cdot |BC|}$$

$$\angle C = \arccos$$

Получив внутренние углы, можно определить остроугольность треугольника – если все углы меньше 90 градусов, то треугольник остроугольный.

Контрольный пример

Рассмотрим треугольник с вершинами (872;-98), (78;7), (78;3):

Вычислим длины сторон:

$$|AB| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} = \sqrt{(78 - 872)^2 + (7 - (-98))^2} = \\ = \sqrt{(-794)^2 + 105^2} = \sqrt{630436 + 11025} = \sqrt{641461} \approx 800.91260447;$$

$$|AC| = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2} = \sqrt{(78 - 872)^2 + (3 - (-98))^2} = \\ = \sqrt{(-794)^2 + 101^2} = \sqrt{630436 + 10201} = \sqrt{640637} \approx 800.39802598;$$

$$|BC| = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2} = \sqrt{(78 - 78)^2 + (3 - 7)^2} = \\ = \sqrt{0^2 + (-4)^2} = \sqrt{0 + 16} = \sqrt{16} = 4.$$

Вычислим внутренние углы по теореме косинусов:

$$\cos \angle A = \frac{|AB|^2 + |AC|^2 - |BC|^2}{2 \cdot |AB| \cdot |AC|} = \frac{(\sqrt{641461})^2 + (\sqrt{640637})^2 - 4^2}{2 \cdot \sqrt{641461} \cdot \sqrt{640637}} = \frac{641461 + 640637 - 16}{2\sqrt{410943650657}} = \frac{1282082}{2\sqrt{410943650657}} = \frac{641041}{\sqrt{410943650657}},$$

$$\angle A = \arccos\left(\frac{641041}{\sqrt{410943650657}}\right) \approx 0.0049544 = \left(0.0049544 \cdot \frac{180}{\pi}\right)^\circ \approx 0.28386625^\circ;$$

$$\cos \angle B = \frac{|AB|^2 + |BC|^2 - |AC|^2}{2 \cdot |AB| \cdot |BC|} = \frac{(\sqrt{641461})^2 + 4^2 - (\sqrt{640637})^2}{2 \cdot \sqrt{641461} \cdot 4} = \frac{641461 + 16 - 640637}{8\sqrt{641461}} = \frac{840}{8\sqrt{641461}} = \frac{105}{\sqrt{641461}},$$

$$\angle B = \arccos\left(\frac{105}{\sqrt{641461}}\right) \approx 1.4393174 = \left(1.4393174 \cdot \frac{180}{\pi}\right)^\circ \approx 82.46681248^\circ;$$

$$\cos \angle C = \frac{|AC|^2 + |BC|^2 - |AB|^2}{2 \cdot |AC| \cdot |BC|} = \frac{(\sqrt{640637})^2 + 4^2 - (\sqrt{641461})^2}{2 \cdot \sqrt{640637} \cdot 4} = \frac{640637 + 16 - 641461}{8\sqrt{640637}} = \frac{-808}{8\sqrt{640637}} = -\frac{101}{\sqrt{640637}},$$

$$\angle C = \arccos\left(-\frac{101}{\sqrt{640637}}\right) \approx 1.69732085 = \left(1.69732085 \cdot \frac{180}{\pi}\right)^\circ \approx 97.24932127^\circ;$$

Организация интерфейса пользователя

Макет ввода/вывода

Fresult1.	formed triples of coordinates:
Fresult2.	d__d.d__d.d__d.d__d__d__d.d__d__d__d.d__d__d__d.d__d__d__d.d__d__d__d
Fresult3.	Task: Find all acute-angled triangles formed by any three points
Fresult4.	number of acute-angled triangles:d__d
Fresult5.	acute-angled triangles:
Fresult6.	d__d (d__d). 1. (d__d;d__d) 2. (d__d;d__d) 3. (d__d;d__d)
Fresult7.	this file is not open or does not exist. please, restart the programm
Fprotocol1.	result file is not open or does not exist. please, restart the programm
Fprotocol2.	this file is not open or does not exist. please, restart the programm
Fprotocol3.	Specified number of points (d__d) has been reduced to N
Fprotocol4.	Readed number of points: d__d
Fprotocol5.	Real number of points: d__d
Fprotocol6.	number of acute-angled triangles: d__d
Fprotocol7.	end of inputFile
Fprotocol8.	there is no paired coordinate for x (d__d.d__d)
Fprotocol9.	d__d.d__d.d__d.d__d__d.d__d__d__d.d__d__d__d.d__d__d__d.d__d__d__d
Fprotocol10.	This is an acute-angled triangle.
Fprotocol11.	This is not an acute-angled triangle.
Fprotocol12.	Selected file:
Cin1.	d
Cin2.	\n
Cout1.	protocol file is not open or does not exist. please, restart the programm

Cout2.	result file is not open or does not exist. please, restart the programm
Cout3.	available files:
Cout4.	d. "ind.txt"
Cout5.	select the file to open :
Cout6.	error: this file is not exist. please, press enter to try again:
Cout7.	this file is not open or does not exist. please, restart the programm
inpurFile1.	s_s
inputFile2.	d__d.d__d

Реализация вывода в консоль

Библиотека	Вывод
iostream	std::cout <<

Внутренний формат хранения данных

Имя	Тип	Назначение
N	const unsigned	Ограничение количества считываемых точек
fresult	fstream	Файл результата
fprotocol	fstream	Файл протокола
P[]	bool	Масиив для хранения истинности остроугольности
inputFile	fstream	Входной файл
x[], y[]	float	Массив считываемых точек
XP[][], YP[][]	long double	Двумерный массив с тройками координат
tryAgain	char	Вспомогательная
file	unsigned	
fileselected	bool	
big_number	int	Необработанное считанное число точек
real_number		Обработанное считанное число точек
number		Реальное число точек
result		Число острых треугольников
index		Число троек координат
i, j , k		Переменные для счета
s	char	Вспомогательная
tmp_x	float	
tmp_y		
longest	int	
x, y ,z		
x1, y1, x2, y2	double	

Работа с файлами

Для работы с файлами используется библиотека «fstream».

Метод	Описание
fstream f	Создание файловой переменной
f.open("file.txt", ios::out)	Открытие файла на запись
f.open("file.txt", ios::in)	Открытие файла на чтение
filein.seekg(-1, ios::cur);	Перемещение курсора назад относительно текущего положения
filein >> noskipws	Чтение пробелов, символов перехода на новую строку
f.is_open()	Определяет, открыт ли файл
f.eof()	Проверка на конец файла
f <<	Запись данных в файл
f >>	Чтение данных из файла

Описание функций

Синтаксис

Имя	Тип возвращаемого значения	Параметры		
		Тип	Имя	Изменение
readFile	int	Fstream	inputFile	&
		Float	x[N], y[N]	-
		int	big_number, real_number, number	&
pointsCreation	void	Float	x[N], y[N]	-
		Long double	XP[100000][3], YP[100000][3]	-
		Int	real_number	-
		Int	index	&
process	int	Long double	XP[100000][3], YP[100000][3]	-
		Int	result	&
		Int	index	-
distance	double	double	x1, y1, x2, y2	-
main	void	-	-	-

Назначение

Имя	Возвращаемое значение	Внешние изменения	Параметры				Назначение
			Входной	Выходной	Модифицируемый	Транзитный	
readFile	0, -1, -2, -3, number	-	inputFile	x[], y[], big_number, real_number, number	fprotocol	-	Считывает координаты из файла, отфильтровывая все лишнее
pointsCreation	-	-	x[], y[]	XP[], YP[], index	fprotocol, fresult	real_number	Создает уникальные тройки координат
process	0	-	XP[], YP[], index	result	fprotocol	index	Проверяет треугольник на остроугольность
distance	$\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$	-	x1, y1, x2, y2	-	-	x1, y1, x2, y2	Высчитывает длины сторон треугольника
main	-	Открытие файлов, резерв	-	-	fprotocol, fresult	-	Основная функция программы

		ЦИЯ ПАМЯТИ ДЛЯ МАССИВ ОВ					
--	--	--------------------------------------	--	--	--	--	--

Описание алгоритма

Программа берет точки и задает треугольники, затем проверяет их на остроугольность. Результат работы программы выводится в файл «result.txt». Все действия программы фиксируются в протоколе «protocol.txt». Блок-схема работы программы представлена ниже:

Текст программы

```
#include <iostream>;
#include <iomanip>;
#include <math.h>;
#include <fstream>;

using namespace std;
const unsigned N = 50;
fstream fresult, fprotocol;
bool P[10000];

int readFile(fstream& inputFile, float x[N], float y[N], int& big_number, int& real_number, int& number);
void pointsCreation(float x[N], float y[N], long double XP[100000][3], long double YP[100000][3], int
real_number, int& index);
int process(long double XP[10000][3], long double YP[10000][3], int& result, int index);
double distance(double x1, double y1, double x2, double y2);
void main(void)
{
    fstream inputFile;
    float x[N], y[N]; long double XP[10000][3], YP[10000][3]; char tryAgain = '!'; unsigned file;
    bool fileSelected = false;
    int big_number = 0, real_number = 0, number = 0, result = 0, index = 0, i, j = 0;
    fresult.open("result.txt", ios::out);
    fprotocol.open("protocol.txt", ios::out);
    if (!fprotocol.is_open())
    {
        cout << "protocol file is not open or does not exist. please, restart the programm";
        return;
    }
    if (!fresult.is_open())
    {
        cout << "result file is not open or does not exist. please, restart the programm";
        fprotocol << "result file is not open or does not exist. please, restart the programm";
        return;
    }
    //file selection
    while (fileSelected != true)
    {
        cout << skipws << "available files:\n";
        cout << "\t0. \"in.txt\"\n";
        cout << "\t1. \"in1.txt\"\n";
        cout << "\t2. \"in2.txt\"\n";
        cout << "\t3. \"in3.txt\"\n";
        cout << "\t4. \"in4.txt\"\n";
        cout << "\t5. \"in5.txt\"\n";
        cout<<"select the file to open : ";
        cin >> file;
        switch (file)
        {
            case 0:
            {
                inputFile.open("in.txt", ios::in);
                fileSelected = true;
                fprotocol << "selected file: in.txt\n";
                break;
            }
            case 1:
```

```

        {
            inputFile.open("in1.txt", ios::in);
            fileSelected = true;
            fprotocol << "selected file: in1.txt\n";
            break;
        }
        case 2:
        {
            inputFile.open("in2.txt", ios::in);
            fileSelected = true;
            fprotocol << "selected file: in2.txt\n";
            break;
        }
        case 3:
        {
            inputFile.open("in3.txt", ios::in);
            fileSelected = true;
            fprotocol << "selected file: in3.txt\n";
            break;
        }
        case 4:
        {
            inputFile.open("in4.txt", ios::in);
            fileSelected = true;
            fprotocol << "selected file: in4.txt\n";
            break;
        }
        case 5:
        {
            inputFile.open("in5.txt", ios::in);
            fileSelected = true;
            fprotocol << "selected file: in5.txt\n";
            break;
        }
        default:
        {
            cout << "\nerror: this file is not exist. please, press enter to try again: " <<
noskipws;
            cin.ignore();
            cin.get();
            system("cls");
            fileSelected = false;
            break;
        }
    }
}
//checking file opening
if (!inputFile.is_open())
{
    cout << "this file is not open or does not exist. please, restart the programm";
    fprotocol << "this file is not open or does not exist. please, restart the programm";
    fresult << "this file is not open or does not exist. please, restart the programm";
    return;
}
//checking number of points
switch (readFile(inputFile, x, y, big_number, real_number, number))
{
    case 0:
    {
        cout << "since the number of points is zero, programm cannot be executed";
        fprotocol << "since the number of points is zero, programm cannot be executed";
        return;
    }
    case -1:
    {
        cout << "the number of points cannot be less than zero";
        fprotocol << "the number of points cannot be less than zero";
        return;
    }
    case -2:
    {
        cout << "file is empty";
        fprotocol << "file is empty";
        return;
    }
    case -3:
    {

```

```

        cout << "file reading error";
        fprotocol << "file reading error";
        return;
    }
}
if (big_number != 0)
{
    //cout << "specified number of points (" << big_number << ") has been reduced to " << N <<
    '\n';
    fprotocol << "specified number of points (" << big_number << ") has been reduced to " << N
    << '\n';
}
//cout << "readed number of points: " << number << '\n';
//cout << "real number of points: " << real_number << '\n';
fprotocol << "readed number of points: " << number << '\n';
fprotocol << "real number of points: " << real_number << '\n';
//process
pointsCreation(x, y, XP, YP, real_number, index);
process(XP, YP, result, index);
//cout << "number of acute-angled triangles:" << result;
fprotocol << "number of acute-angled triangles:" << result;
fresult << "Task: Find all acute-angled triangles formed by any three points\n";
fresult << "number of acute-angled triangles:" << result << '\n';
fresult << "acute-angled triangles:\n";
for (i = 0; i < index; i++)
{
    if (P[i] == true)
    {
        j++;
        fresult << setw(2) << left << j << " (" << setw(3) << i << ")." << right << "\t1."
        << '(' << setw(7) << left << XP[i][0] << ";" << setw(7) << right << YP[i][0] << ')' << "\t2." << '(' <<
        setw(7) << left << XP[i][1] << ";" << setw(7) << right << YP[i][1] << ')' << "\t3." << '(' << setw(7) <<
        left << XP[i][2] << ";" << setw(7) << right << YP[i][2] << ')' << endl;
    }
}
inputFile.close();
}
int readFile(fstream& inputFile, float x[N], float y[N], int& big_number, int& real_number, int& number)
{
    int i = 0; char s = '!'; float tmp_x, tmp_y;
    inputFile >> number;
    //checking number of points
    if (inputFile.eof())
    {
        return -2;
    }
    else if (number < 0)
    {
        return -1;
    }
    else if (number == 0)
    {
        return 0;
    }
    //programm will work
    else
    {
        if (number > N)
        {
            big_number = number;
            number = N;
        }
        while(i < number)
        {
            inputFile << skipws;
            inputFile >> tmp_x;
            if (inputFile.eof())
            {
                fprotocol << "end of inputFile\n";
                return number;
            }
            else
            {
                inputFile << noskipws;
                do inputFile >> s;
                while (s == ' ' || s == '\t');
                if (s == '\n')

```

```

        {
            //реакция на то, что есть только x
            fprotocol << "there is no paired coordinate for x (" << tmp_x <<
            "\n";
            continue;
        }
        else if (inputFile.eof())
        {
            //реакция на то, что есть только x
            fprotocol << "there is no paired coordinate for x (" << tmp_x <<
            "\n";
            return number;;
        }
        else
        {
            inputFile.seekg(-1, ios::cur);
            s = '!';
            inputFile << skipws;
            inputFile >> tmp_y;
            if (inputFile.eof())
            {
                //реакция на конец файла
                fprotocol << "end of inputFile\n";
                return number;
            }
            else
            {
                x[i] = tmp_x;
                y[i] = tmp_y;
                //cout << i+1 << ".\t" << x[i] << ' ' << y[i] << '\n';
                fprotocol << i + 1 << ".\t" << x[i] << ' ' << y[i] <<
                '\n';

                i++;
                real_number = i;
                inputFile << noskipws;
                while (!inputFile.eof() && s != '\n') inputFile >> s;
                s = '!';
            }
        }
    }
    }
    return number;
}
return -3;
}

void pointsCreation(float x[N], float y[N], long double XP[100000][3], long double YP[100000][3], int
real_number, int& index)
{
    int i, j, k;
    for (i = 0; i < real_number; i++)
    {
        for (j = i + 1; j < real_number; j++)
        {
            for (k = j + 1; k < real_number; k++)
            {
                XP[index][0] = x[i];
                XP[index][1] = x[j];
                XP[index][2] = x[k];
                YP[index][0] = y[i];
                YP[index][1] = y[j];
                YP[index][2] = y[k];
                index++;
            }
        }
    }
    //cout << endl;
    fresult << "formed triples of coordinates:\n";
    for (i = 0; i < index; i++) {
        //cout << i+1 << ".\t" << XP[i][0] << " " << YP[i][0] << "\t\t" << XP[i][1] << " " <<
        YP[i][1] << "\t\t" << XP[i][2] << " " << YP[i][2] << endl;
        fresult << i + 1 << ".\t" << XP[i][0] << " " << YP[i][0] << "\t\t" << XP[i][1] << " " <<
        YP[i][1] << "\t\t" << XP[i][2] << " " << YP[i][2] << endl;
        fprotocol << i + 1 << ".\t" << XP[i][0] << " " << YP[i][0] << "\t\t" << XP[i][1] << " " <<
        YP[i][1] << "\t\t" << XP[i][2] << " " << YP[i][2] << endl;
    }
    return;
}
}

```

```

int process(long double XP[10000][3], long double YP[10000][3], int& result, int index)
{
    int longest, x, y, z, i;

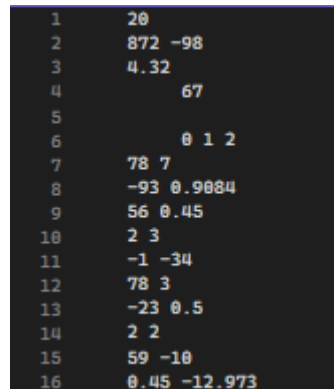
    for (i = 0; i < index; i++)
    {
        x = distance(XP[i][0], YP[i][0], XP[i][1], YP[i][1]);
        y = distance(XP[i][1], YP[i][1], XP[i][2], YP[i][2]);
        z = distance(XP[i][2], YP[i][2], XP[i][0], YP[i][0]);
        longest = z;
        if (longest < x) {
            z = longest;
            longest = x;
            x = z;
        }
        if (longest < y) {
            z = longest;
            longest = y;
            y = z;
        }
        if (x * x + y * y > longest * longest) {
            //cout << i + 1 << ".\t" << "This is an acute-angled triangle.\n";
            fprotocol << i + 1 << ".\t" << "This is an acute-angled triangle.\n";
            P[i] = true;
            result++;
        }
        else
        {
            //cout << i + 1 << ".\t" << "This is not an acute-angled triangle.\n";
            fprotocol << i + 1 << ".\t" << "This is not an acute-angled triangle.\n";
            P[i] = false;
        }
    }
    return 0;
}

double distance(double x1, double y1, double x2, double y2)
{
    return sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
}

```

Вывод программы

Контрольный пример: входной файл (рис. 5), файл с результатом (рис. 6), протокол (рис. 7).



```

1      20
2      872 -98
3      4.32
4          67
5
6          0 1 2
7      78 7
8      -93 0.9084
9      56 0.45
10     2 3
11     -1 -34
12     78 3
13     -23 0.5
14     2 2
15     59 -10
16     0.45 -12.973

```

Рисунок 5.

```

1  number of acute-angled triangles:39
2  acute-angled triangles:
3  1 (14) . 1.(872) . -98) 2.(78) . 7) 3.(78) . 3)
4  2 (34) . 1.(872) . -98) 2.(2) . 3) 3.(-1) . -34)
5  3 (37) . 1.(872) . -98) 2.(2) . 3) 3.(2) . 2)
6  4 (39) . 1.(872) . -98) 2.(2) . 3) 3.(0.45) . -12.973)
7  5 (42) . 1.(872) . -98) 2.(-1) . -34) 3.(2) . 2)
8  6 (44) . 1.(872) . -98) 2.(-1) . -34) 3.(0.45) . -12.973)
9  7 (53) . 1.(872) . -98) 2.(2) . 2) 3.(0.45) . -12.973)
10 8 (59) . 1.(0) . 1) 2.(78) . 7) 3.(78) . 3)
11 9 (66) . 1.(0) . 1) 2.(-93) . 0.9084) 3.(-1) . -34)
12 10 (73) . 1.(0) . 1) 2.(56) . 0.45) 3.(-1) . -34)
13 11 (78) . 1.(0) . 1) 2.(56) . 0.45) 3.(0.45) . -12.973)
14 12 (82) . 1.(0) . 1) 2.(2) . 3) 3.(2) . 2)
15 13 (86) . 1.(0) . 1) 2.(-1) . -34) 3.(-23) . 0.5)
16 14 (88) . 1.(0) . 1) 2.(-1) . -34) 3.(59) . -10)
17 15 (93) . 1.(0) . 1) 2.(78) . 3) 3.(0.45) . -12.973)
18 16 (103) . 1.(78) . 7) 2.(-93) . 0.9084) 3.(78) . 3)
19 17 (110) . 1.(78) . 7) 2.(56) . 0.45) 3.(78) . 3)
20 18 (116) . 1.(78) . 7) 2.(2) . 3) 3.(78) . 3)
21 19 (118) . 1.(78) . 7) 2.(2) . 3) 3.(2) . 2)
22 20 (126) . 1.(78) . 7) 2.(78) . 3) 3.(-23) . 0.5)
23 21 (127) . 1.(78) . 7) 2.(78) . 3) 3.(2) . 2)
24 22 (143) . 1.(-93) . 0.9084) 2.(2) . 3) 3.(-1) . -34)
25 23 (146) . 1.(-93) . 0.9084) 2.(2) . 3) 3.(2) . 2)
26 24 (148) . 1.(-93) . 0.9084) 2.(2) . 3) 3.(0.45) . -12.973)
27 25 (151) . 1.(-93) . 0.9084) 2.(-1) . -34) 3.(2) . 2)
28 26 (162) . 1.(-93) . 0.9084) 2.(2) . 2) 3.(0.45) . -12.973)
29 27 (167) . 1.(56) . 0.45) 2.(2) . 3) 3.(2) . 2)
30 28 (177) . 1.(56) . 0.45) 2.(78) . 3) 3.(59) . -10)
31 29 (186) . 1.(2) . 3) 2.(-1) . -34) 3.(-23) . 0.5)
32 30 (188) . 1.(2) . 3) 2.(-1) . -34) 3.(59) . -10)
33 31 (191) . 1.(2) . 3) 2.(78) . 3) 3.(2) . 2)
34 32 (194) . 1.(2) . 3) 2.(-23) . 0.5) 3.(2) . 2)
35 33 (196) . 1.(2) . 3) 2.(-23) . 0.5) 3.(0.45) . -12.973)
36 34 (197) . 1.(2) . 3) 2.(2) . 2) 3.(59) . -10)
37 35 (199) . 1.(2) . 3) 2.(59) . -10) 3.(0.45) . -12.973)
38 36 (204) . 1.(-1) . -34) 2.(-23) . 0.5) 3.(2) . 2)
39 37 (207) . 1.(-1) . -34) 2.(2) . 2) 3.(59) . -10)
40 38 (217) . 1.(-23) . 0.5) 2.(2) . 2) 3.(0.45) . -12.973)
41 39 (219) . 1.(2) . 2) 2.(59) . -10) 3.(0.45) . -12.973)

```

Рисунок 6.

```

selected file: in.txt
1. 872 -98
there is no paired coordinate for x (4.32)
there is no paired coordinate for x (67)
2. 0 1
3. 78 7
4. -93 0.9084
5. 56 0.45
6. 2 3
7. -1 -34
8. 78 3
9. -23 0.5
10. 2 2
11. 59 -10
12. 0.45 -12.973
end of inputFile
readed number of points: 20
real number of points: 12
1. 872 -98 0 1 78 7
2. 872 -98 0 1 -93 0.9084
3. 872 -98 0 1 56 0.45
4. 872 -98 0 1 2 3
5. 872 -98 0 1 -1 -34
6. 872 -98 0 1 78 3
7. 872 -98 0 1 -23 0.5
8. 872 -98 0 1 2 2
9. 872 -98 0 1 59 -10
...
219. -23 0.5 59 -10 0.45 -12.973
220. 2 2 59 -10 0.45 -12.973
1. This is not an acute-angled triangle.
2. This is not an acute-angled triangle.
3. This is not an acute-angled triangle.
4. This is not an acute-angled triangle.
5. This is not an acute-angled triangle.
6. This is not an acute-angled triangle.
7. This is not an acute-angled triangle.
8. This is not an acute-angled triangle.
9. This is not an acute-angled triangle.
...
219. This is not an acute-angled triangle.
220. This is an acute-angled triangle.
number of acute-angled triangles:39

```

Рисунок 7.

Выводы

В ходе данной курсовой работы были рассмотрены следующие понятия: обработка и хранение числовых данных, файловые переменные, режимы открытия и закрытия файла, проверка на то, открыт ли файл, считывание и запись информации в файл, динамическая память, указатели, арифметика указателей, удаление данных, одномерные, двумерные и трёхмерные массивы, функции, параметры функций, возвращаемое значение функций, ссылки, типы данных, условные конструкции, циклы и функция `main`. Также на практике была реализована программа, решающая геометрическую задачу на пересечение фигур на координатной плоскости.