**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

Курсовая РАБОТА

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1302 |  | Новиков Г.В. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2021

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студент Новиков Г.В. | | |
| Группа 1302 | | |
|  | | |
| Исходные данные:  Рассматривая точки пересечения прямых, проходящих через любую пару из заданных N точек на плоскости, определить принадлежность кругу радиуса R: в) с центром в точках из N заданных максимального количества точек пересечения. | | |
| Содержание пояснительной записки:  Исходная формулировка, особенности задания, контрольный пример, формальная постановка задания, формат хранения данных, ограничения, макет ввода и вывода, средства обеспечения ввода и вывода, алгоритм решения, программа, результаты, выводы, список используемых материалов. | | |
|  | | |
| Дата выдачи задания: 23.11.2021 | | |
| Дата сдачи: 17.12.2021 | | |
| Дата защиты: 17.12.2021 | | |
| Студент |  | Новиков Г.В. |
| Преподаватель |  | Калмычков В.А. |

**Аннотация**

В курсовую работу входит программа, которая обрабатывает и корректирует двумерный массив из файла и выполняет индивидуальное задание. Программа выполняется по средствам некоторых математических формул, таких как: построение прямых по координатам заданных точек, нахождение точек пересечения прямых, проверка на принадлежность точек окружности радиуса R с заданным центром. В результате программа выводит в файл откорректированный двумерный массив, основные действия программы и ответ на заданное индивидуальное задание.

**Summary**

The course work includes a program that processes and corrects a two-dimensional array from a file and performs an individual task. The program is executed by means of some mathematical formulas, such as: plotting straight lines from the coordinates of given points, finding lines intersection points, points belonging to circle with radius R and given center check. As a result, the program outputs to the file the corrected two-dimensional array, the main actions of the program and the answer to the given individual task.

Оглавление

[Исходная формулировка 5](#_Toc90620857)

[Особенности задания 5](#_Toc90620858)

[Контрольный пример 5](#_Toc90620859)

[Формальная постановка задания 5](#_Toc90620860)

[Формат хранения данных 5](#_Toc90620861)

[Ограничения, обусловленные выполнением на компьютере 6](#_Toc90620862)

[Макет ввода и вывода 6](#_Toc90620863)

[Средства обеспечения ввода и вывода 7](#_Toc90620864)

[Алгоритм решения 8](#_Toc90620865)

[Программа 11](#_Toc90620866)

[Результаты работы программы 14](#_Toc90620867)

[Выводы о проделанной работе 15](#_Toc90620868)

[Список использованных источников 16](#_Toc90620869)

# Исходная формулировка

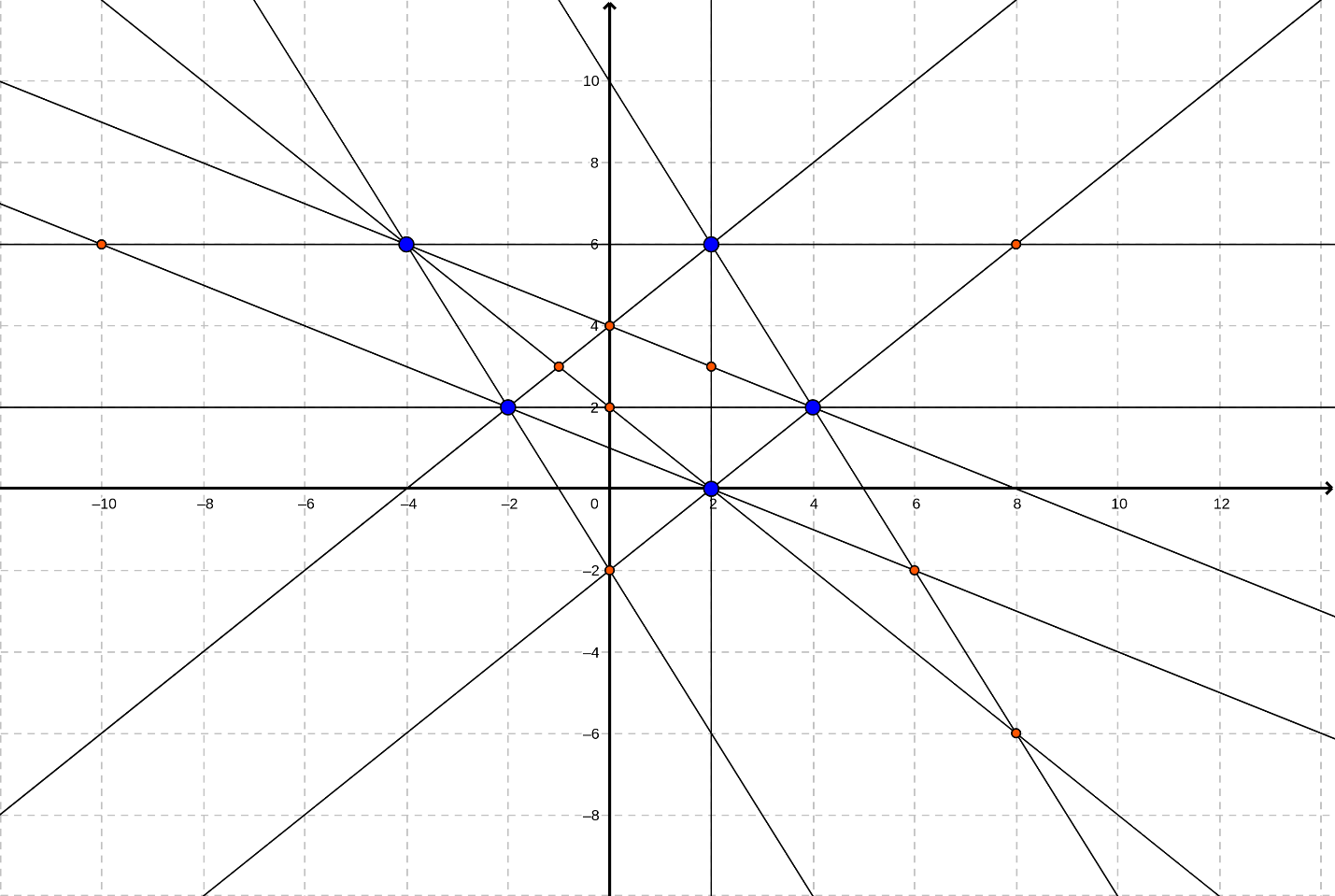
Рассматривая точки пересечения прямых, проходящих через любую пару из заданных N точек на плоскости, определить принадлежность кругу радиуса R: в) с центром в точках из N заданных максимального количества точек пересечения.

# Особенности задания

Обработка двумерного массива с использованием функций и с организацией файлового ввода/вывода.

# Контрольный пример

In.txt:



# Формальная постановка задания

Дано:

Найти: прямоугольный треугольник, в который входит больше всего точек

Способ решения: обработка массива с применением функций

# Формат хранения данных

|  |  |  |
| --- | --- | --- |
| Идентификатор типа | Тип | Назначение |
| N | const int | Размер массива (максимальный) |
| a | int | Размер массива |
| i, j, k, p | int | Индексная переменная |
| W[N][N], A[N][N] | int | Массив |
| d | double | Временная переменная |
| tmp | int | Временная переменная |
| f, f1, f2 | fstream | Файл |
| x, x1, x2, xp, y, y1, y2, yp | double | Переменные для координат точек |
| a, b, c | double | Переменные для сторон треугольника |
| s | double | Площадь треугольника |
| s | char | Переменная для обработки |
| ap, bp, cp | double | Переменные для проверки треугольника |
| eps | double | eps = 1e-8 |
| z | double | Переменная для обработки сторон треугольника |
| most, maxcount | double | Переменные для подсчета точек, входящих в треугольник |
| n, tr | int | Переменные для номеров треугольников |
| outarr | void | Функции вывода массива |
| inp | int | Функция ввода массива |
| triangle | void | Функция обработки массива |
| main | int | Функция main |

# Ограничения, обусловленные выполнением на компьютере

Тип double предоставляет доступ к вещественным числам в диапазоне от +/- 1.7E-308 до 1.7E+308.

# Макет ввода и вывода

|  |  |
| --- | --- |
| Приветствие | Романова Ольга Валерьевна \nГруппа: 1302\nНачало:04.12.21\nКонец: 16.12.21 \nКурсовая работа \nФормулировка: Даны N точек на плоскости. Найти среди них точки, являющиеся вершинами фигуры, содержащей максимальное число заданных точек.Фигура: д)прямоугольный треугольник.\n |
| Номер треугольника | f1 << "Номер Треугольника: " << n << endl; |
| Координаты вершин | f1 << "Точка A: " << x << ' ' << y << endl;  f1 << "Точка B: " << x1 << ' ' << y1 << endl;  f1 << "Точка C: " << x2 << ' ' << y2 << endl; |
| Стороны треугольника | f1 << "Сторона АВ (катет) = " << a << endl;  f1 << "Сторона АC (катет) = " << b << endl;  f1 << "Сторона BC (гипотенуза) = " << c << endl; |
| Проверка треугольника | f1 << "Не треугольник\n"; |
| Проверка прямоугольного треугольника | f1 << "Треугольник прямоугольный и его площадь: " << s << "\n";  f1 << "Треугольник не прямоугольный \n"; |
| Точка | f1 << "Точка P: " << xp << ' ' << yp << endl; |
| Принадлежность точки треугольнику | f1 << "Точка принадлежит треугольнику\n";  f1 << "Точка не принадлежит\n";  f1 << setw(50) << "Треугольник: " << n << " содержит " << maxcount << " точек\n"; |
| Коррекция размера | f1 << "Изначальный размер: " << a << '\n';  f1 << "Размер, скорректирован в " << a << '\n';  f1 << "Размер, скорректирован в " << a << '\n';  f1 << "Итоговый массив: \n"; |
| Результат | f1 << "Треугольник " << tr << " содержит больше всего точек: " << most;  f2 << "Треугольник " << tr << " содержит больше всего точек: " << most; |

# Средства обеспечения ввода и вывода

Средство обеспечения ввода:

f>>

Средство обеспечения вывода в консоль:

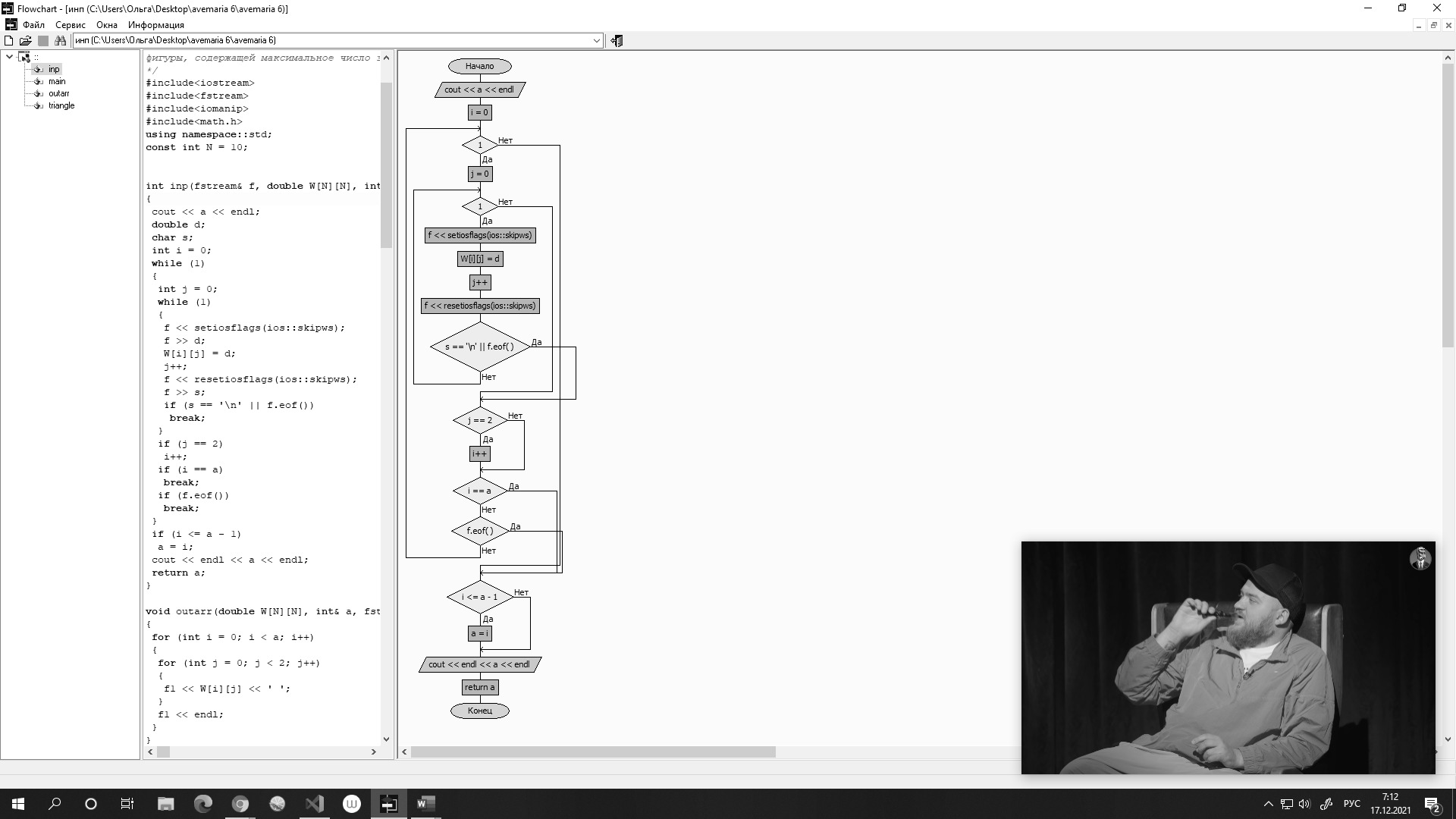
cout <<

Средство обеспечения вывода в файл:

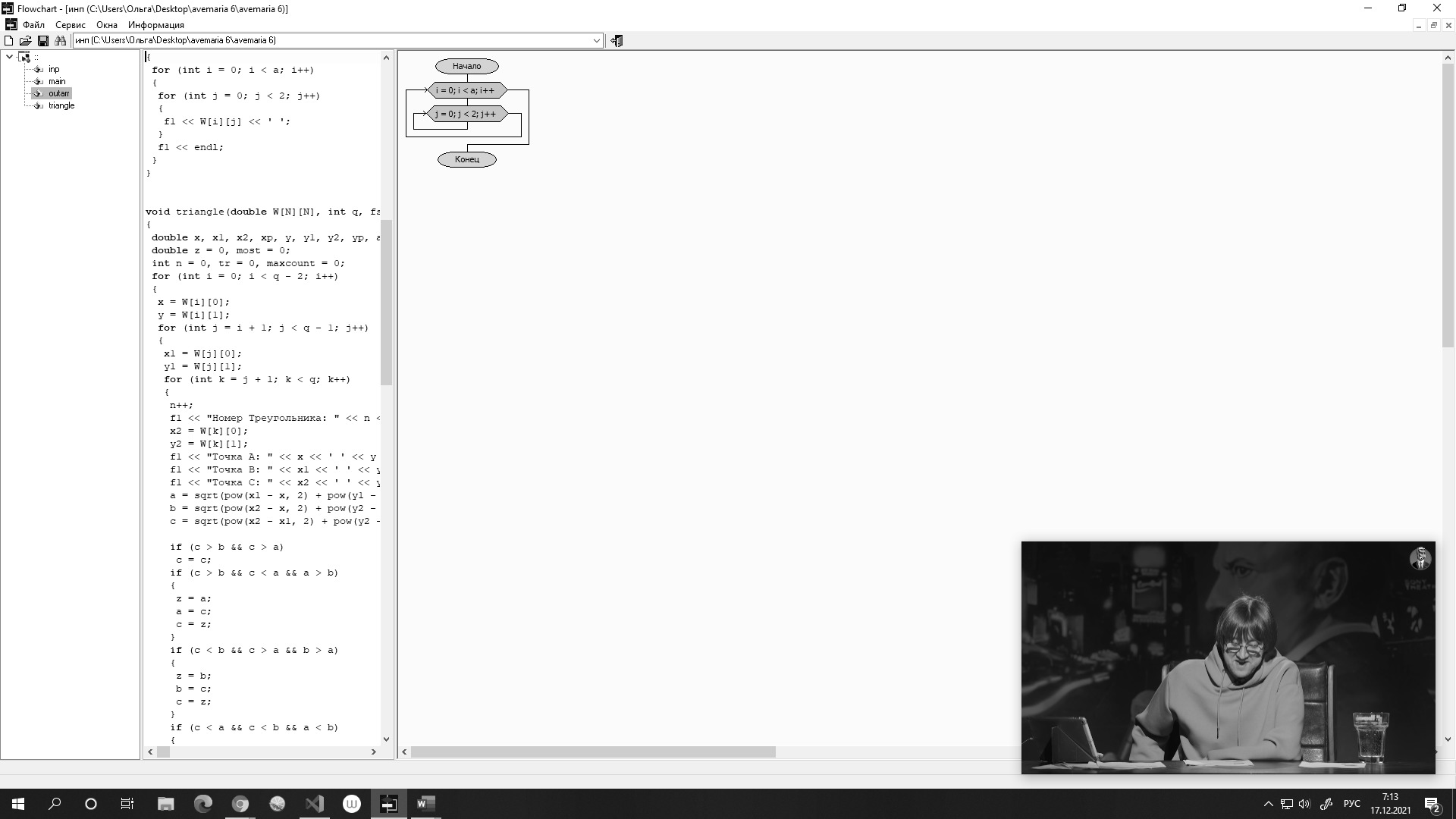
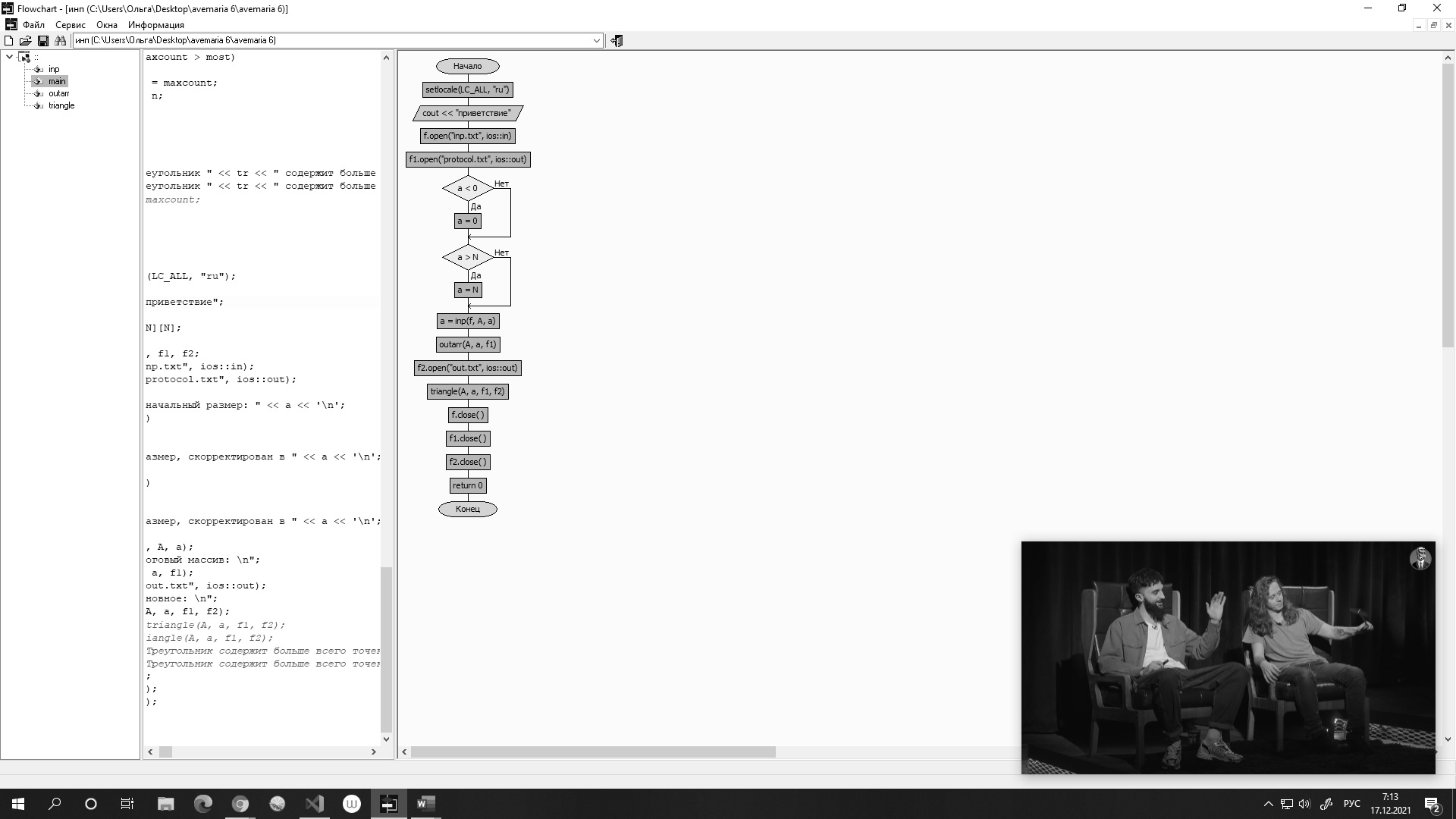
f1<<

f2<<

# Алгоритм решения

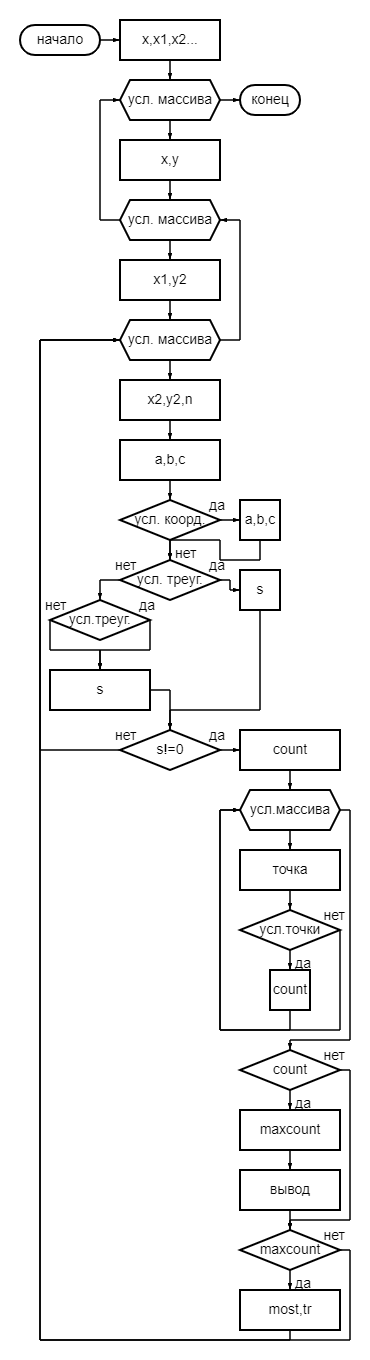


Функция inp



Функция main

Функция ourarr



Функция triangle

# Программа

/\*ФИО: Романова Ольга Валерьевна

\*Группа: 1302

\*Дата начала:04.12.21

\*Дата окончания:16.12.21

\*Курсовая работа

\*Формулировка: Даны N точек на плоскости. Найти среди них точки, являющиеся вершинами

фигуры, содержащей максимальное число заданных точек. Фигура: д)прямоугольный треугольник.

\*/

#include<iostream>

#include<fstream>

#include<iomanip>

#include<math.h>

using namespace::std;

const int N = 10;

int inp(fstream& f, double W[N][N], int& a)

{

cout << a << endl;

double d;

char s;

int i = 0;

while (1)

{

int j = 0;

while (1)

{

f << setiosflags(ios::skipws);

f >> d;

W[i][j] = d;

j++;

f << resetiosflags(ios::skipws);

f >> s;

if (s == '\n' || f.eof())

break;

}

if (j == 2)

i++;

if (i == a)

break;

if (f.eof())

break;

}

if (i <= a - 1)

a = i;

cout << endl << a << endl;

return a;

}

void outarr(double W[N][N], int& a, fstream& f1)

{

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

{

for (int j = 0; j < 2; j++)

{

f1 << W[i][j] << ' ';

}

f1 << endl;

}

}

void triangle(double W[N][N], int q, fstream& f1, fstream& f2)

{

double x, x1, x2, xp, y, y1, y2, yp, a, b, c, s = 0, ap, bp, cp, eps = 1e-8;

double z = 0, most = 0;

int n = 0, tr = 0, maxcount = 0;

for (int i = 0; i < q - 2; i++)

{

x = W[i][0];

y = W[i][1];

for (int j = i + 1; j < q - 1; j++)

{

x1 = W[j][0];

y1 = W[j][1];

for (int k = j + 1; k < q; k++)

{

n++;

f1 << "Номер Треугольника: " << n << endl;

x2 = W[k][0];

y2 = W[k][1];

f1 << "Точка A: " << x << ' ' << y << endl;

f1 << "Точка B: " << x1 << ' ' << y1 << endl;

f1 << "Точка C: " << x2 << ' ' << y2 << endl;

a = sqrt(pow(x1 - x, 2) + pow(y1 - y, 2));

b = sqrt(pow(x2 - x, 2) + pow(y2 - y, 2));

c = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

if (c > b && c > a)

c = c;

if (c > b && c < a && a > b)

{

z = a;

a = c;

c = z;

}

if (c < b && c > a && b > a)

{

z = b;

b = c;

c = z;

}

if (c < a && c < b && a < b)

{

z = b;

b = c;

c = z;

}

if (c < b && c < a && b < a)

{

z = a;

a = c;

c = z;

}

f1 << "Сторона АВ (катет) = " << a << endl;

f1 << "Сторона АC (катет) = " << b << endl;

f1 << "Сторона BC (гипотенуза) = " << c << endl;

if (((x - x1) / (x2 - x1) == (y - y1) / (y2 - y1)) || (a == 0 || b == 0 || c == 0))

{

f1 << "Не треугольник\n";

s = 0;

}

else

{

if (((a \* a) + (b \* b) == (c \* c)) || (abs((c \* c) - ((a \* a) + (b \* b))) < eps))

{

s = (a \* b) / 2;

f1 << "Треугольник прямоугольный и его площадь: " << s << "\n";

}

else

{

s = 0;

f1 << "Треугольник не прямоугольный \n";

}

}

if (s != 0)

{

int count = 0;

for (int p = 0; p < q; p++)

{

xp = W[p][0];

yp = W[p][1];

f1 << "Точка P: " << xp << ' ' << yp << endl;

ap = (x - xp) \* (y1 - y) - (x1 - x) \* (y - yp);

bp = (x1 - xp) \* (y2 - y1) - (x2 - x1) \* (y1 - yp);

cp = (x2 - xp) \* (y - y2) - (x - x2) \* (y2 - yp);

if ((ap < 0 && bp < 0 && cp < 0) || (ap > 0 && bp > 0 && cp > 0) || (ap == 0 || bp == 0 || cp == 0))

{

count++;

f1 << "Точка принадлежит треугольнику\n";

}

else

f1 << "Точка не принадлежит\n";

}

if (count > maxcount)

{

maxcount = count;

f1 << setw(50) << "Треугольник: " << n << " содержит " << maxcount << " точек\n";

}

if (maxcount > most)

{

most = maxcount;

tr = n;

}

}

}

}

}

f1 << "Треугольник " << tr << " содержит больше всего точек: " << most;

f2 << "Треугольник " << tr << " содержит больше всего точек: " << most;

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Романова Ольга Валерьевна \nГруппа: 1302\nНачало:04.12.21\nКонец: 16.12.21 \nКурсовая работа \nФормулировка: Даны N точек на плоскости. Найти среди них точки, являющиеся вершинами фигуры, содержащей максимальное число заданных точек.Фигура: д)прямоугольный треугольник.\n";

double A[N][N];

int a;

fstream f, f1, f2;

f.open("inp.txt", ios::in);

f1.open("protocol.txt", ios::out);

f >> a;

f1 << "Изначальный размер: " << a << '\n';

if (a < 0)

{

a = 0;

f1 << "Размер, скорректирован в " << a << '\n';

}

if (a > N)

{

a = N;

f1 << "Размер, скорректирован в " << a << '\n';

}

a = inp(f, A, a);

f1 << "Итоговый массив: \n";

outarr(A, a, f1);

f2.open("out.txt", ios::out);

triangle(A, a, f1, f2);

f.close();

f1.close();

f2.close();

return 0;

}

# Результаты работы программы

Вывод в консоль

Подробный вывод в файл

Входной файл и краткий вывод

# Выводы о проделанной работе

Работа с вводом и выводом из текстового файла и обработка двумерного массива с помощью функций.

# Список использованных источников

1. Описание методических указаний / сост.: Калмычков В.А. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2021.

2. Описание составления отчета по курсовой работе / СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2021.