**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

Курсовая РАБОТА

**по дисциплине «Программирование»**

Тема: Использование массивов для решения геометрической задачи.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1309 |  | Ищенко Д.О. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2021

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студентка Ищенко Д.О. | | |
| Группа 1309 | | |
| Тема работы: Использование массивов для решения геометрической задачи | | |
| Исходные данные:  Дан набор координат точек на плоскости. Найти среди них точки, являющиеся вершинами фигуры, содержащей наибольшее количество заданных точек. Фигура представляет собой прямоугольник с полуокружностями на противоположных друг другу сторонах (эти стороны образуют диаметры полуокружностей). Полуокружности располагаются таким образом, что площадь всей фигуры больше площади исходного прямоугольника. | | |
| Содержание пояснительной записки:  Содержание, введение, заключение, список использованных источников.  Предполагаемый объем пояснительной записки:  Не менее 5 страниц. | | |
| Дата выдачи задания: 16.11.2021 | | |
| Дата сдачи реферата: 22.12.2021 | | |
| Дата защиты реферата: 22.12.2021 | | |
| Студентка |  | Ищенко Д.О. |
| Преподаватель |  | Калмычков В.А. |

**Аннотация**

Задача нахождения фигуры решается перебором всех возможных комбинаций пяти вершин и проверкой каждой комбинации на выполнение набора условий. Этот набор позволяет однозначно определить, образуют ли данные точки искомую фигуру. Задача нахождения количества точек, принадлежащих данной фигуре представляет собой две подзадачи: о принадлежности точки окружности и о принадлежности точки прямоугольнику. Первая подзадача решается проверкой выполнения нестрогого неравенства уравнения окружности, вторая – с использованием утверждения о равенстве площади прямоугольника и суммы площадей треугольников, образованных соединением вершин прямоугольника с точкой, находящейся внутри него и его сторонами.

В ходе написания программы были задействованы структуры, как способ внутреннего хранения данных. Не смотря на это, написанная в ходе выполнения курсовой работы программа не является объектно-ориентированной и использует функции как основной способ разделения логики решения глобальной задачи.

Все взаимодействие с пользователем происходит через три файла: файл входных данных, файл выходных данных, файл протокола программы.

**Summary**

The problem of finding a figure is solved by enumerating all possible combinations of five vertices and checking each combination for the fulfillment of a set of conditions. This set allows you to unambiguously determine whether the given points form the desired shape. The problem of finding the number of points belonging to a given figure consists of two subproblems: the belonging of a point to a circle and the belonging of a point to a rectangle. The first subproblem is solved by checking the fulfillment of the non-strict inequality of the equation of the circle, the second - using the statement about the equality of the area of a rectangle and the sum of the areas of triangles formed by connecting the vertices of a rectangle with a point located inside it and its sides.

In the course of writing the program, structures were involved as a way of internal data storage. Despite this, the program written in the course of the course work is not object-oriented and uses functions as the main way to separate the logic for solving a global problem.

All interaction with the user occurs through three files: the input data file, the output data file, the program protocol file.

**содержание**

[Введение 5](#_Toc90979861)

[1. Анализ задания 6](#_Toc90979862)

[1.1. Формулировка задания. 6](#_Toc90979863)

[1.2. Выявление неточностей. 6](#_Toc90979864)

[1.3. Контрольный пример 8](#_Toc90979865)

[2. Описание решения задачи 10](#_Toc90979866)

[2.1. Математическая постановка задачи. 10](#_Toc90979867)

[2.2. Формат внутреннего представления данных. 12](#_Toc90979868)

[2.3. Описание программной реализации решения. 13](#_Toc90979869)

[3. Интерфейсы взаимодействия с пользователем 13](#_Toc90979870)

[3.1. Описание формата файлов. 13](#_Toc90979871)

[3.2. Обработка некорректных входных данных. 13](#_Toc90979872)

[4. Текст программы 15](#_Toc90979873)

[5. Результат работы программы 15](#_Toc90979874)

[5.1. Случай 1. Нехватка точек. 15](#_Toc90979875)

[список использованных источников 17](#_Toc90979876)

[Приложение А 19](#_Toc90979877)

формат внутреннего представления данных (рисунки работы с динамической памятью, четко понимать какие указатели за что отвечают), описание работы функций (табличкой, имя функций, назначение, параметры, внешние эффекты), описание алгоритма решения (на каждую функцию своя блок-схема, пояснение алгоритмов), текст программы, результат работы программы(входные и выходные файлы, поясняющие картинки – что произошло), вывод.

# 

# Введение

Кратко описать цель работы, основные задачи им методы их решения.

# Анализ задания

## Формулировка задания.

Дан набор координат точек на плоскости. Найти среди них точки, являющиеся вершинами фигуры, содержащей наибольшее количество заданных точек. Фигура представляет собой прямоугольник с полуокружностями на противоположных друг другу сторонах (эти стороны образуют диаметры полуокружностей). Полуокружности располагаются таким образом, что площадь всей фигуры больше площади исходного прямоугольника (рис. 1).

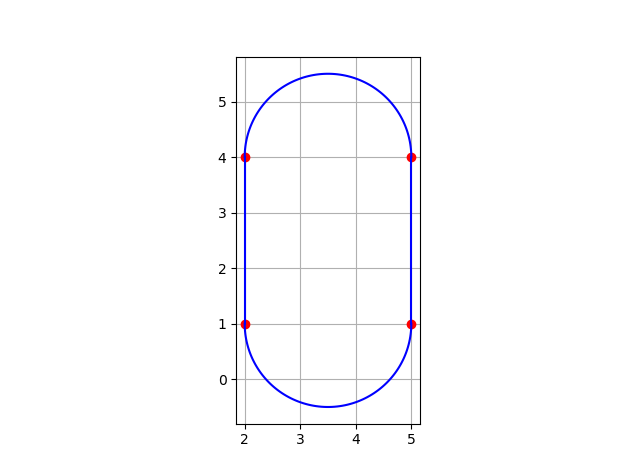


рис.

## Выявление неточностей.

Фигура, представленная в задании, образовывается в общем случае четырьмя точками. Но при этом невозможно однозначно интерпретировать её вид, имея только эти четыре точки (на рис.1 и рис.2 изображены две фигуры, имеющие различный вид, но одинаковые координаты точек).

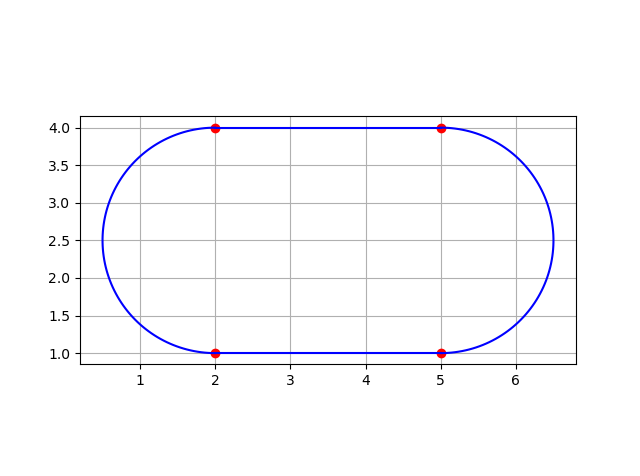


рис.

Поэтому для однозначной интерпретации фигуры будем брать ещё одну точку из набора данных, которая должна представлять собой центр одной из полуокружностей и лежать на середине одной из сторон прямоугольника (рис. 3).

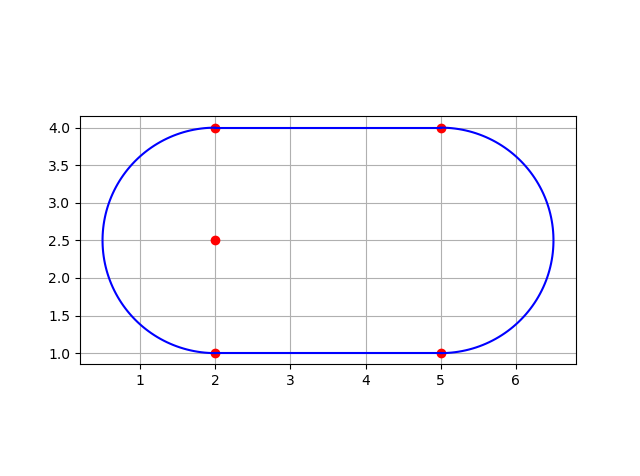


рис.

Ещё одним неоднозначным моментом является вопрос о включении в множество принадлежащих фигуре точек тех, которые лежат на периметре фигуры. В данной работе, такие точки учитываются, при подсчете.

Также данную фигуру можно рассматривать, как прямоугольник и две окружности (а не полуокружности). В таком случае подзадача проверки принадлежности точки фигуре сильно упрощается, а смысл задания в целом остается прежним. Тогда фигура с рис. 3 примет вид фигуры на рис. 4.

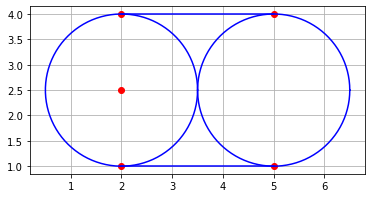


рис.

## Контрольный пример

Дан набор точек (рис. 5):

{ (6.0:8.0), (-4.0:-5.0), (-4.0:0.0), (5.0:5.0), (0.5:5.0), (5.0:-15.0), (5.0:-15.0), (5.0:-5.0), (9.0:1.0), (12.0:3.0), (-4.0:5.0), (-5.0:-5.0), (-5.0:-5.0), (1.0:0.0), (0.0:1.0), (5.0:7.0), (5.0:-3.0), (2.0:-8.0), (9.0:0.0), (9.0:0.0), (0.0:0.0), (2.0:2.0), (5.0:-5.0), (4.0:4.0), (9.0:-1.0), (9.0:-1.0), (9.0:-1.0), (9.0:-1.0)}

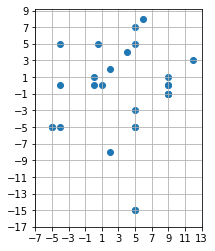


рис.

Для данного набора можно построить три фигуры, удовлетворяющие заданию.

Фигура 1 имеет вершины в точках {(5 : 5), (0.5 : 5), (-4 : 5), (-4 : -5), (5 : -5)} (рис. 6) и включает в себя 13 точек.

Фигура 2 имеет вершины в точках {(-4 : -5), (-4 : 0), (-4 : 5), (5 : 5), (5 : -5)} (рис. 7) и включает в себя 15 точек.

Фигура 3 имеет вершины в точках {(0 : 0), (2 : 2), (4 : 4), (9 : -1), (5 : -5)} (рис. 8) и включает в себя 8 точек.

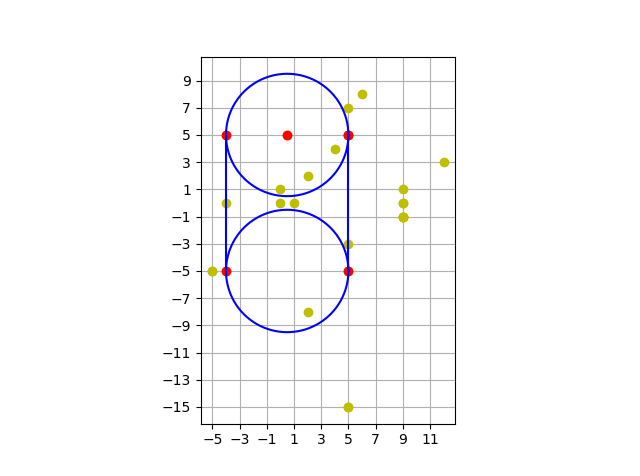


рис.

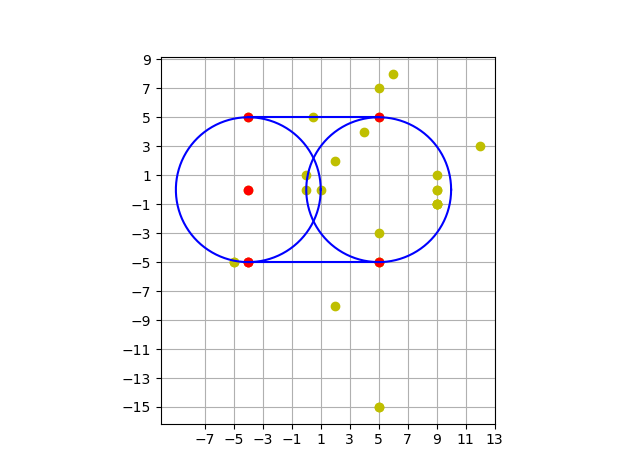


рис.

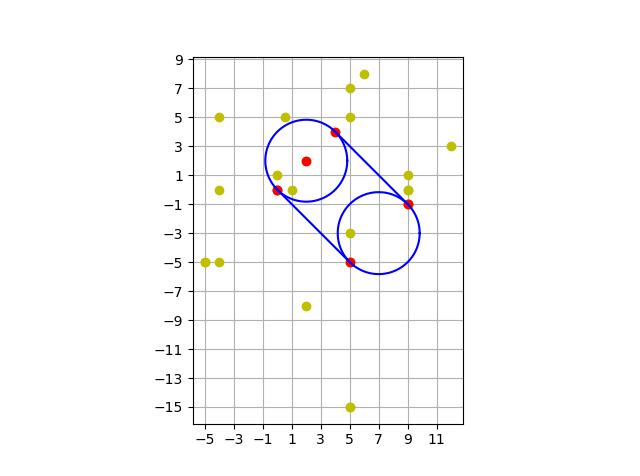


рис.

Наибольшее количество точек в себя включает фигура 2, поэтому вершины этой фигуры и будут ответом на задание.

# Описание решения задачи

## Математическая постановка решения.

*Дано:* xi,yi 0 <= i < n

*Найти:* координаты фигуры, включающей в себя наибольшее количество точек

Искомая фигура представляет собой прямоугольник, на середине одной из сторон которого лежит одна из точек. Эту фигуру задают пять точек.

*Решение:*

Множество из пяти координат точек (xi,yi) образует искомую фигуру, когда:

1. В нем есть три коллинеарные точки (лежащие на одной прямой).
2. Эти три точки образуют отрезок, разделенный одной из точек пополам (отрезок AC, где B принадлежит AC, и AB=BC).
3. Все пять точек являются различными (не совпадают).
4. Оставшиеся две точки (D, E) образуют отрезок, равный по длине AC, и CD = AE
5. CD перпендикулярен DE

Последний пункт можно проверить так:

Найдем координаты точки пересечения диагоналей предполагаемого прямоугольника S (пусть будут sx и sy).

sx = (ax+dx)/2 {1}

sy = (ay+dy)/2

где (ax, ay) и (dx,dy) это координаты точек A и B соответственно. Тогда, если **AS = CS = SD** можно построить окружность с радиусом AS. И точка С будет образовывать с точками A и D угол, опирающийся на диаметр окружности (рис. 9). По теореме Фалеса об угле, опирающемся на диаметр окружности угол DCA прямой. Учитывая это и пункт 4) можно утверждать, что ACDE – прямоугольник. А точка B лежит на середине стороны AC и точки A, B, C, D, E – образуют искомую фигуру.

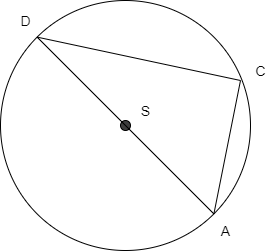


рис.

Чтобы понять, принадлежит точка фигуре, нужно проверить, принадлежит ли она хотя бы одной из окружностей фигуры или прямоугольнику.

Проверить принадлежность окружности можно по уравнению окружности:

где (x,y) – координаты точки. Для первой окружности r = AB, (x0,y0) = (bx,by). А для второй необходимо найди координаты центра второй окружности, аналогично {1} (fx,fy). Тогда r = DF, (x0,y0) = (fx,fy).

Проверить принадлежность некоторой точки К прямоугольнику можно выполнением равенства S1+S2+S3+S4 = S. Где S = площадь ACDE, S1 – площадь ACK, S2 – площадь CKD, S3 – площадь DKE, S4 – площадь EKA (рис. 10).

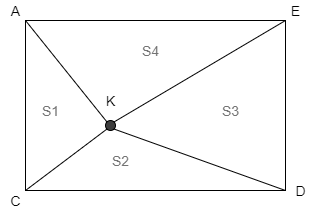


рис.

Зная координаты вершин, площадь треугольника можно найти по формуле Герона, предварительно вычислив длины его сторон.

Проверив таким образом каждую точку из данного набора, можно найти количество точек, принадлежащих фигуре. Повторив это для каждой найденной фигуры, найдем ту, для которой это количество будет максимальным. Координаты её вершины будут являться ответом на задание.

## Формат внутреннего представления данных.

Для хранения множества координат точек в программе используется динамически выделенный массив с названием points пользовательского типа coord. Структура coord включает в себя два публичных поля: double x, double y. Размер массива n определяется функцией get\_n (см. описание в подразделе 3.2) и фактически означает количество строк, в начале которых через пробел находятся два числа. При этом, поскольку считывание элементов не происходит, невозможно найти дубликаты вершин. Это означает, что фактический размер используемой памяти может быть меньше выделенного. На окончание массива будет указывать (points + n).

Для удобства работы с найденными вершинами фигур вводится ещё один пользовательский тип – fig. Он содержит публичные поля:

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Значение по умолчанию | Назначение |
| first | coord | {0,0} | Координаты первой точки (А в мат. постановке задачи). |
| second | coord | {0,0} | Координаты второй точки (B) – данного центра одной из окружностей. |
| third | coord | {0,0} | Координаты третьей точки (С). |
| four | coord | - | Координаты четвертой точки (D). |
| five | coord | - | Координаты пятой точки (E). |
| center | coord | - | Координаты центра окружности (точка пересечения диагоналей прямоугольника) (S). |
| sub\_center | coord | - | Координаты центра второй окружности (F). |
| inner\_points\_count | int | 5 | Количество точек, принадлежащих фигуре. |

**Про значения по умолчанию**. Координаты первой и второй точек фигуры имеют одинаковые значения по умолчанию, это используется для проверки ненахождения фигуры с помощью функции is\_empty. Поскольку точно известно, что ни одна вершина фигуры не может совпадать с другой, равенство первых двух точек будет указывать на то, что значение по умолчанию не было изменено, а значит, фигура не найдена. Счетчик же inner\_points\_count равен по умолчанию пяти, так как для любой фигуры минимум пять точек, её образующих, будут ей принадлежать. (при подсчете не учитываются точки center и sub\_center, так как они не входят в исходный набор точек, а являются вспомогательными.

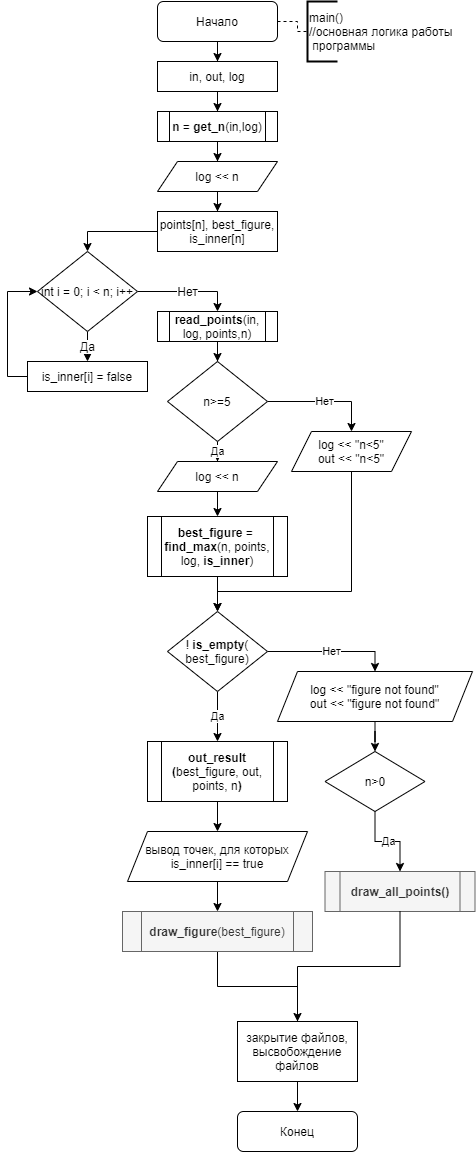
Также для определения набора точек, принадлежащих фигуре, используется динамический массив is\_inner типа bool размерностью n. Значение для i ячейки этого массива определяет, принадлежит ли i точка из массива points фигуре.

## Описание программной реализации решения.

Далее приведены описания и блок-схемы всех функций программы (кроме двух – draw\_all\_points и draw\_figure, их описание см. в приложении А). Начнем описание программной реализации с функции **main.**

В таблице представлены все локальные переменных функции main.

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| in | fstream | Файл входных данных |
| out | fstream | Файл выходных данных |
| log | fstream | Файл протокола |
| n | int | Количество значащих элементов в массиве points и is\_inner |
| VISUALIZATE | bool | Указывает функции find\_max вызывать функцию draw\_figure для каждой найденной фигуры (см. Приложение А) |
| points | coord\* | Указатель на начало динамического массива всех считанных точек |
| best\_figure | fig | Данные фигуры, отвечающей условию задачи |
| is\_inner | bool\* | Указатель на начало динамического массива-маски для массива points. Отражает, входит ли каждая конкретная точка в фигуру best\_figure. |



Далее следует описание функций, в порядке их вызова в функции main, кроме функций get\_n и read\_point и всех, вызываемых ими (их описание см. в подразделе 3.2)

Функция **find\_max** вызывает функции distance, is\_equal, is\_collinear, find\_inner\_points, log\_fig\_coords. Их описание в см. таблице ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Аргументы | Возвращаемое значение | Внешние эффекты | Назначение |
| Find\_max | int n, coord\* points, std::fstream& log,  bool\* is\_inner\_mask | Наилучшая найденная фигура типа fig. | Вывод значений в файл протокола, изменяет массив is\_inner mask | Находит среди точек points фигуры, выводит их в файл протокола, считает количество внутренних точек, выбирает фигуру с наибольшим количеством внутренних точек и возвращает её. |
| is\_equal | coord& a, coord& b | bool | - | Сравнивает координаты двух точек или сравнивает координаты одной точки и первых трех значений фигуры. |
| fig& figure, coord& p |
| is\_collinear | coord& a, coord& b, coord& c | bool | - | Проверяет, лежат ли переданные три точки на одной прямой. |
| log\_fig coords | fig figure, fstream& log | - | Вывод в файл протокола | Выводит данные фигуры в файл протокола. |
| find inner points | coord\* points, fig & figure, int n, bool \*is\_inner |  | Изменяет массив is\_inner mask |  |
| distance | coord a, coord b | double | - | Возвращает расстояние между точками а и b |

# Интерфейсы взаимодействия с пользователем

## Описание формата файлов.

Выходной файл *output.txt* содержит:

1. Все считанные точки
2. Координаты искомой фигуры
3. Координаты точек, принадлежащих этой фигуре.

Файл протокола *log.txt* содержит:

1. Сообщения о некорректных строках во входном файле
2. Количество строк, значения которых будут считаны
3. Список всех считанных координат точек
4. Сообщения о найденных дубликатах
5. Реально считанное количество точек n
6. Координаты всех найденных фигур и количество принадлежащих им точек.

Файл входных данных *input.txt* содержит набор координат xi, yi (по два числа в каждой строке).

## Обработка некорректных входных данных.

Обработкой считывания данных в программе занимаются 4 функции check\_line, get\_n, is\_in, read\_points.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Назначение | Параметры | Внешние эффекты | Возвращаемое значение |
| check\_line | Проверяет переданную строку на наличие в ней двух чисел в начале, разделенных пробелом | std::string l | - | Логическое значение (bool) |
| get\_n | Считает количество строк во входном файле, удовлетворяющих check\_line. Выводит ошибки и номера строк в файл протокола. | std::fstream& in, std::fstream& log | Возможен вывод в файл протокола | Количество строк (int) |
| read\_points | Считывает значения точек, проверяет повторы, выводит предупреждение о повторах в файл протокола, выводит считанные значения в файл протокола, корректирует в соответствии с эти значение n. Формирует вспомогательный выходной файл *beauty\_points.txt* для работы вспомогательной программы (см. приложение А) | std::fstream& in, coord\* points, int &n, std::fstream &log | Изменяет n, изменяет значения массива points, выводит информацию в файл протокола, выводит информацию во вспомогательный файл. | - |
| is\_in | Проверяет вхождение координат точки в массив. | coord point, coord\* points, const int n | - | Логическое значение (bool) |

Блок-схемы

Данная логика работы позволяет обрабатывать некорректный ввод, например:

1. Пустые строки входного файла игнорируются.
2. Строки, не содержащие в начале два числа через пробел, игнорируются.
3. Строки, содержащие два числа через пробел и ненужную информацию после этого, обрабатываются. Числа считываются, остальное отбрасывается.

Реальное количество считанных данных может быть меньше, выделенного изначально размера массива из-за встречающихся дубликатов, при этом размерность массива n корректируется и не позволяет выйти за пределы заполненной части массива.

# Текст программы

# Результат работы программы

## Случай 1. Нехватка точек.

**заключение**

Кратко подвести итоги, проанализировать соответствие поставленной цели и полученного результата.

# список использованных источников

***Ниже представлены примеры библиографического описания, В качестве названия источника в примерах приводится вариант, в котором применяется то или иное библиографическое описание.***

1. Иванов И. И. Книга одного-трех авторов. М.: Издательство, 2010. 000 с.

2. Книга четырех авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров, В. В. Васильев. СПб.: Издательство, 2010. 000 с.

3. Книга пяти и более авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров и др.. СПб.: Издательство, 2010. 000 с.

4. Описание книги под редакцией / под ред. И.И. Иванова СПб., Издательство, 2010. 000 с.

5. Иванов И.И. Описание учебного пособия и текста лекций: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.

6. Описание методических указаний / сост.: И.И. Иванов, П.П. Петров. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.

7. Иванов И.И. Описание статьи с одним-тремя авторами из журнала // Название журнала. 2010, вып. (№) 00. С. 000–000.

8. Описание статьи с четырьмя и более авторами из журнала / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название журнала. 2010, вып. (№) 00. С. 000–000.

9. Иванов И.И. Описание тезисов доклада с одним-тремя авторами / Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПБ, 2010, С. 000–000.

10. Описание тезисов доклада с четырьмя и более авторами / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПБ, 2010, С. 000–000.

11. Описание электронного ресурса // Наименование сайта. URL: http://east-front.narod.ru/memo/latchford.htm (дата обращения: 00.00.2010).

12. ГОСТ 0.0–00. Описание стандартов. М.: Изд-во стандартов, 2010.

13. Пат. RU 00000000. Описание патентных документов / И. И. Иванов, П. П. Петров, С. С. Сидоров. Опубл. 00.00.2010. Бюл. № 00.

14. Иванов И.И. Описание авторефератов диссертаций: автореф. дисс. канд. техн. наук / СПбГЭТУ «ЛЭТИ», СПБ, 2010.

15. Описание федерального закона: Федер. закон [принят Гос. Думой 00.00.2010] // Собрание законодательств РФ. 2010. № 00. Ст. 00. С. 000–000.

16. Описание федерального постановления: постановление Правительства Рос. Федерации от 00.00.2010 № 00000 // Опубликовавшее издание. 2010. № 0. С. 000–000.

17. Описание указа: указ Президента РФ от 00.00.2010 № 00 // Опубликовавшее издание. 2010. № 0. С. 000–000.

# Приложение А

**Вспомогательная программа визуализации фигуры на языке python**