МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

Лабораторная работа №5

**«Разветвляющиеся алгоритмические структуры.**

**Программная реализация базовых разветвляющихся**

**структур и типовых алгоритмов»**

**по дисциплине**

**«Алгоритмизация и программирование»**

Выполнил: студент гр. БЭИ2202 Кулешов А. С.

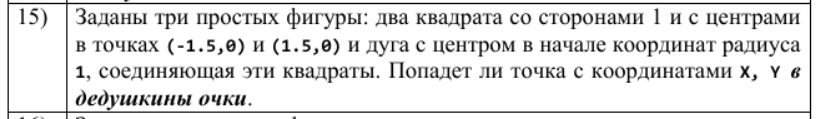
Вариант №15

Проверил: доц. Воробейчиков Л. А.

Москва, 2022 г

1. Индивидуальное задание.

Индивидуальное задание.



1. Формализация и уточнение задания

Требуемая Фигура:

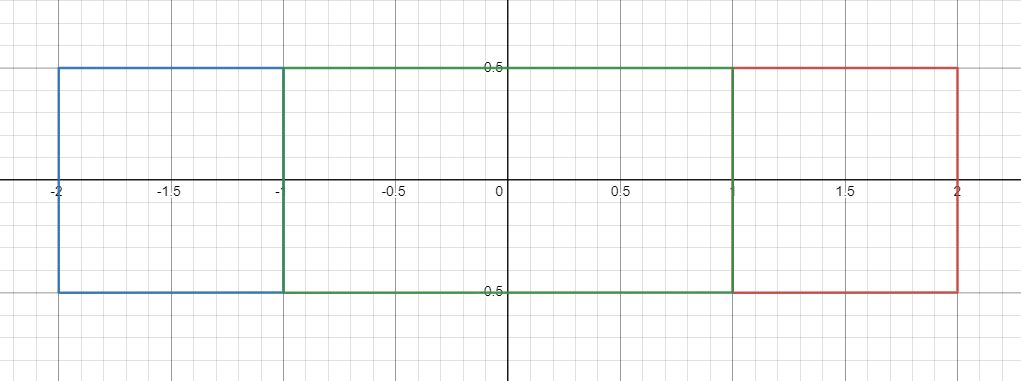


Рисунок 1 – необходимая фигура

1. Т.к. рисование фигуры, параллельно с её проектированием происходило в онлайн-платформе “desmos”, то я могу использовать формулу данной фигура оттуда

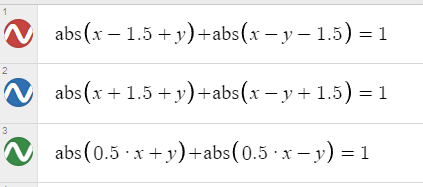


Рисунок 2 – Формулы, задающие «Дедушкины Очки»

1. Начнём проектирование алгоритма методом «сверху-вниз». Т.к. конечная формула слишком длинная, разобью её на под формулы (будем проверять на нахождение в каждой фигуре отдельно).

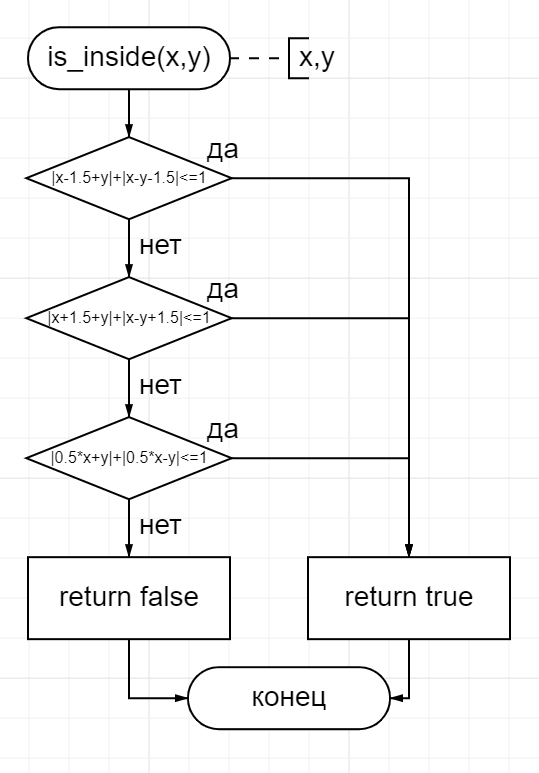


Рисунок 3 – схема функции is\_inside

Можно также заметить, что «очки» на самом деле представляют собой лишь 1 прямоугольник, так что формулу можно сократить, и представить гораздо проще. Представлю новую схему

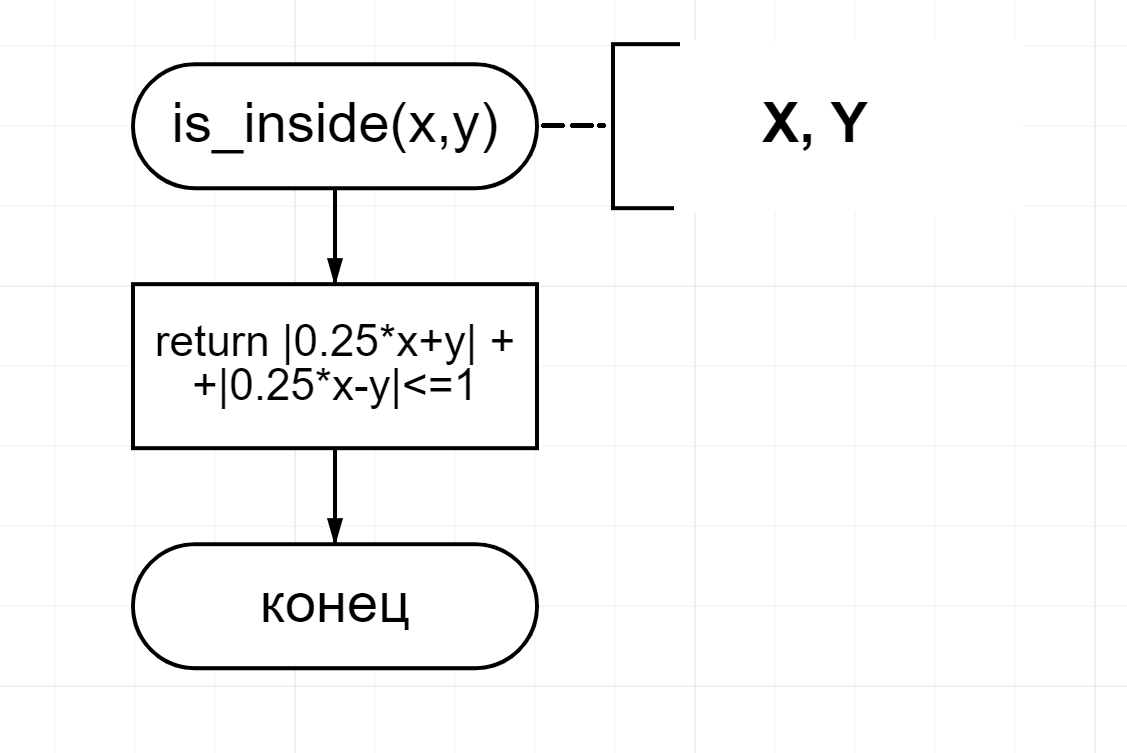


Рисунок 4 – схема упрощённой функции is\_inside

Действительно, в графическом редакторе также видно, как эта формула обобщает «Дедушкины очки». Теперь напишу код, который реализует эту программу.

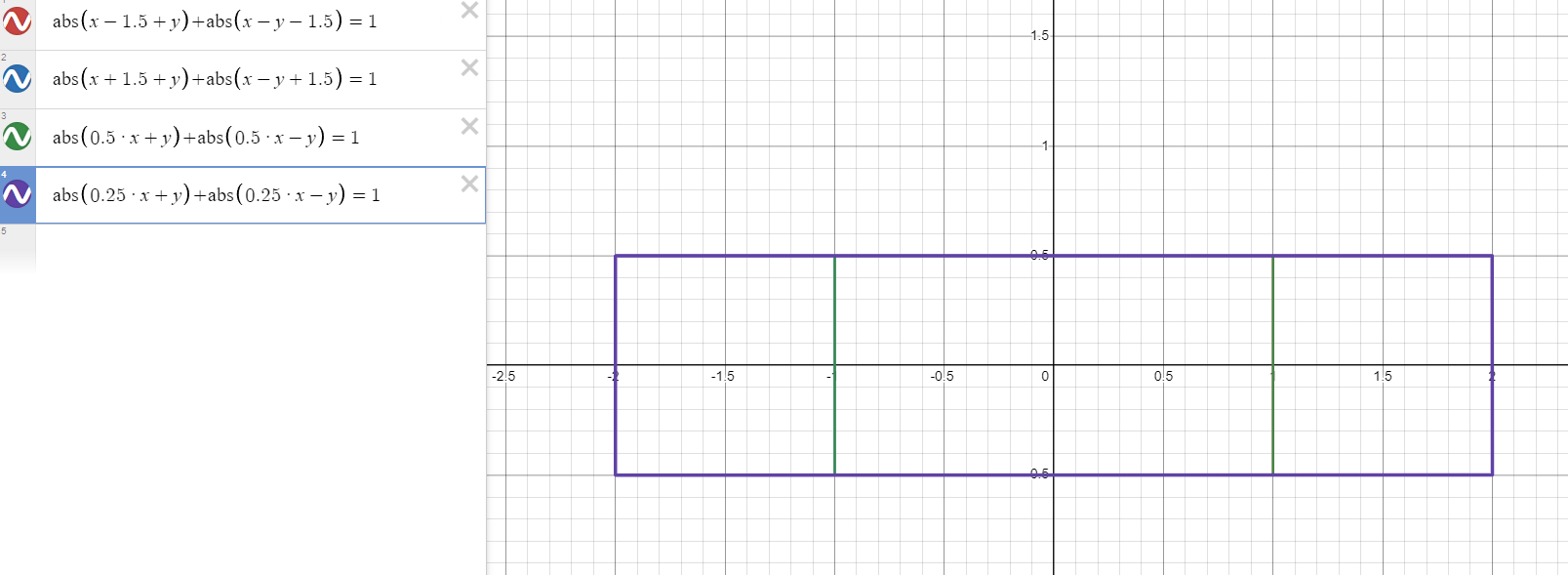


Рисунок 5 – упрощение фигуры

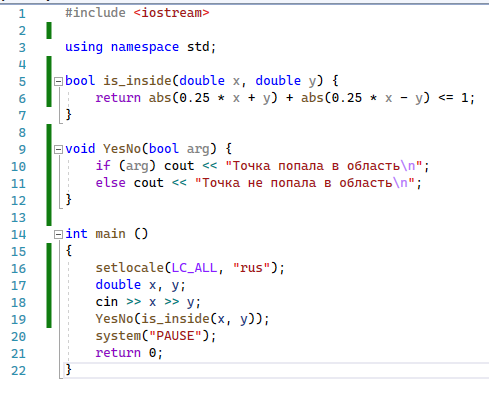


Рисунок 6 – Программный код решения.

Также проверю алгоритм на разных тестовых данных.



Рисунок 7 – Тестовые данные (1; 0.5)



Рисунок 8 – Тестовые данные (2; 1)

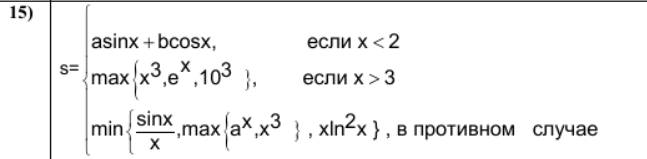


Рисунок 9 – Тестовые данные (1.3; -0.5)

Алгоритм работает корректно на тестовых данных. Результат ручного просчёта совпадает с работой алгоритма

Задание 2.

Индивидуальное задание:



**Формализация и уточнение задания**

Алгоритм решения данной задачи представляет собой комбинацию вычисления сложного выражения с условием и выбора наименьшего (наибольшего) из нескольких значений, используя все виды разветвлений. Можно решить эту задачу двумя способами:

* создать функциональный алгоритм и соответствующую программную функцию, используя вложенные разветвления с базовыми алгоритмами нахождения наибольшего и наименьшего значений, без использования библиотечных или собственных функций max и min;
* создать функциональный алгоритм и соответствующую программную функцию, используя вложенные разветвления и собственные алгоритмы, и программные функции нахождения наибольшего и наименьшего из двух значений. Будем считать, что исходные данные и результат вычислений имеют тип double. Для контроля правильности результата создадим переменную целого типа, этой переменной будем присваивать номер ветви разветвления, по которой выполнялись вычисления.

**Разработка схем алгоритмов**

Схема алгоритма calc, использующего библиотечные алгоритмы нахождения наибольшего и наименьшего значения

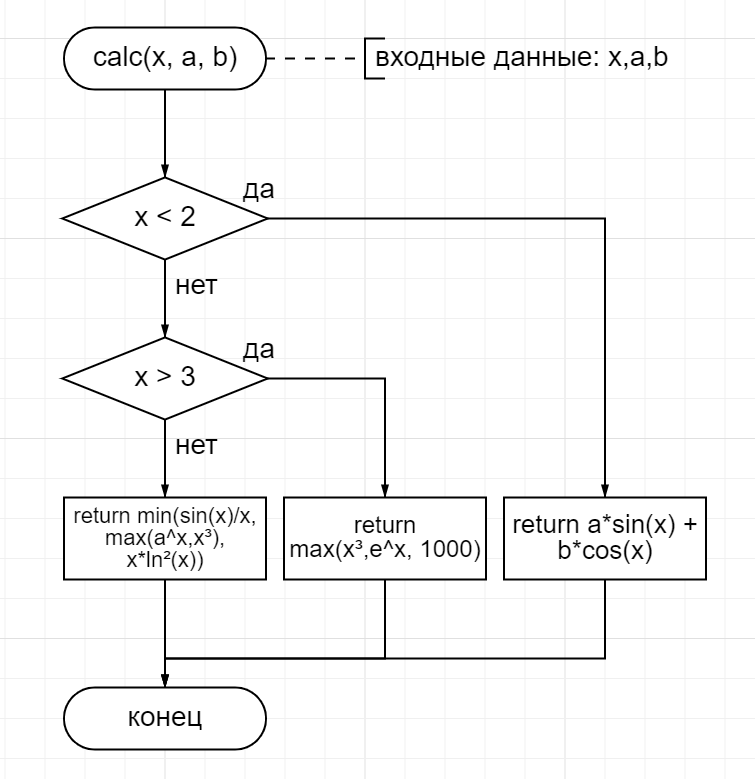


Рисунок 10 – Схема алгоритма calc

Также сделаю схему, использующую ветвление.

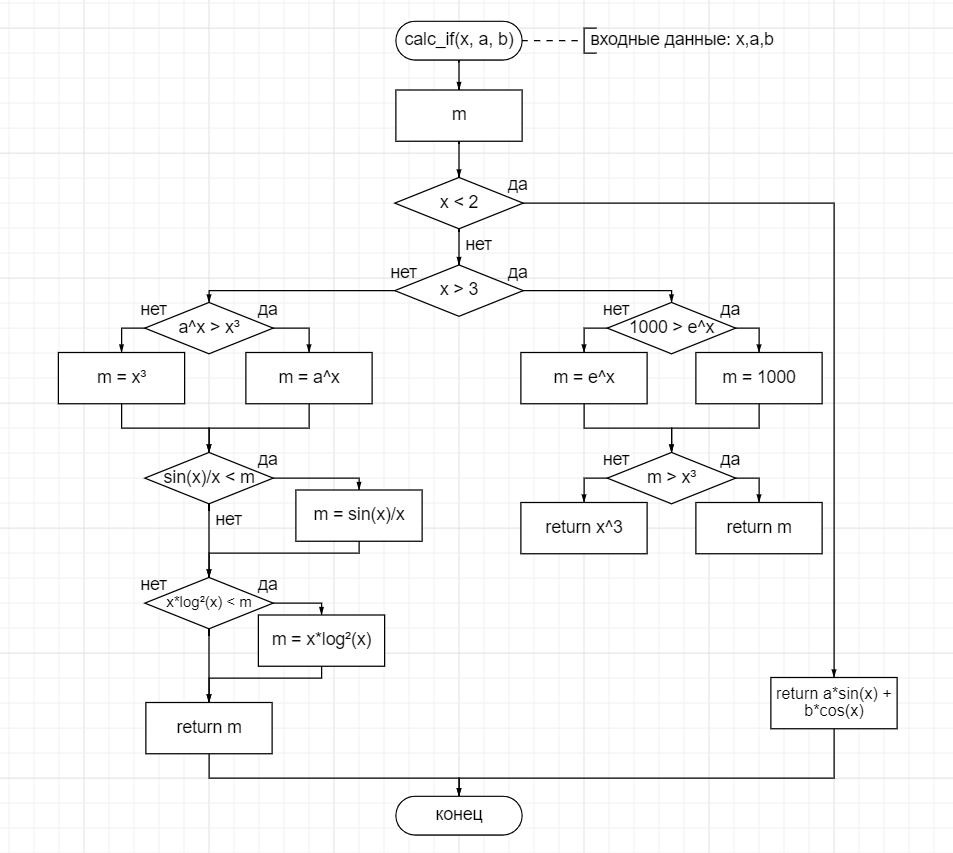


Рисунок 11 – Схема алгоритма с ветвлением calc\_if

Теперь реализую данный алгоритм в виде кода.

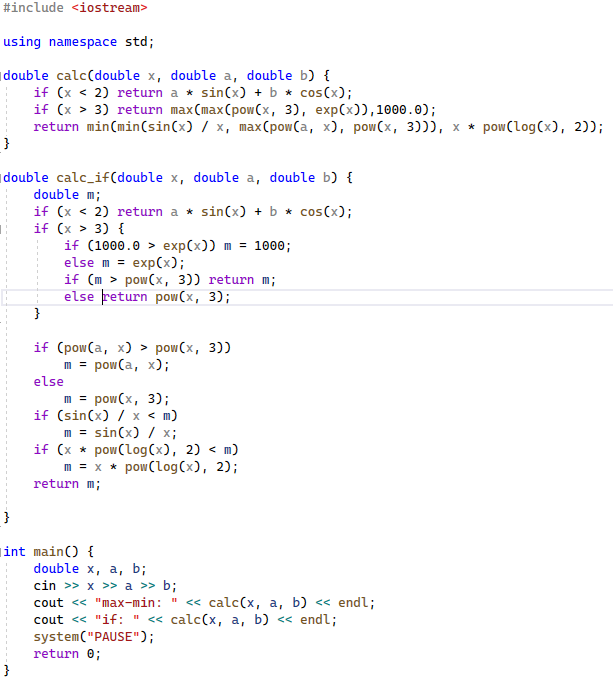


Рисунок 12 – Программный код для задания.

Наконец, проверю код тестовыми данными



Рисунок 13 – Тестовые данные (1; 2; 3)



Рисунок 14 – Тестовые данные (2; 3; 4)



Рисунок 15 – Тестовые данные (3; 3; 3)

Как можно заметить оба алгоритма дают одинаковый результат, что неудивительно, ведь они выполняют одинаковый функционал

**Доказательство правильности результата**

При тестовых данных результаты ручного расчета и вычисления на компьютере совпадают.