«СИМВОЛИЧЕСКИЙ ОБРАЗ АТТРАКТОРА ХЕНОНА И ЛОКАЗИЗАЦИЯ ЦЕПНО-РЕКУРРЕНТНОГО МНОЖЕСТВА»

В ходе проделанной работыбыла поставлена задача изучения построения цепно-рекуррентного множества символических образов.

Для такого построения было необходимо решить задачу поиска сильно связных компонент в графе.

Для проверки работы алгоритма было взято отображение Хенона:

при а = 1.4 и b = 0.3

**Компонентой сильной связности** называется такое (максимальное по включению) подмножество вершин, что любые две вершины этого подмножества достижимы друг из друга, т.е. для :

**Областью сильной связности** называется множество вершин компоненты сильной связности.

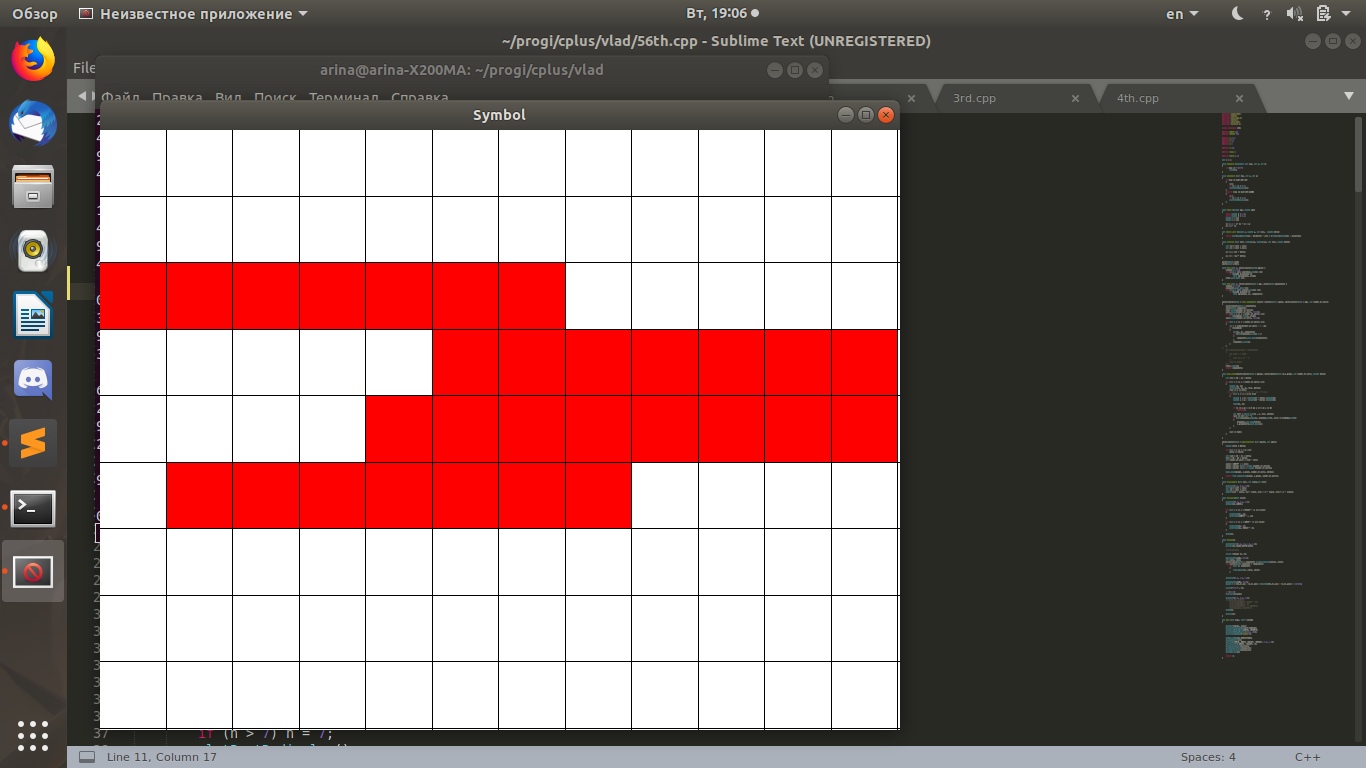
Для нахождения таких компонент был использован алгоритм Косараю, который работает за линейное время благодаря двойному поиску в глубину: одному – в графе и второму в графе , где - состоит из вершин и ребер, направление которых было изменено на обратное, графа .

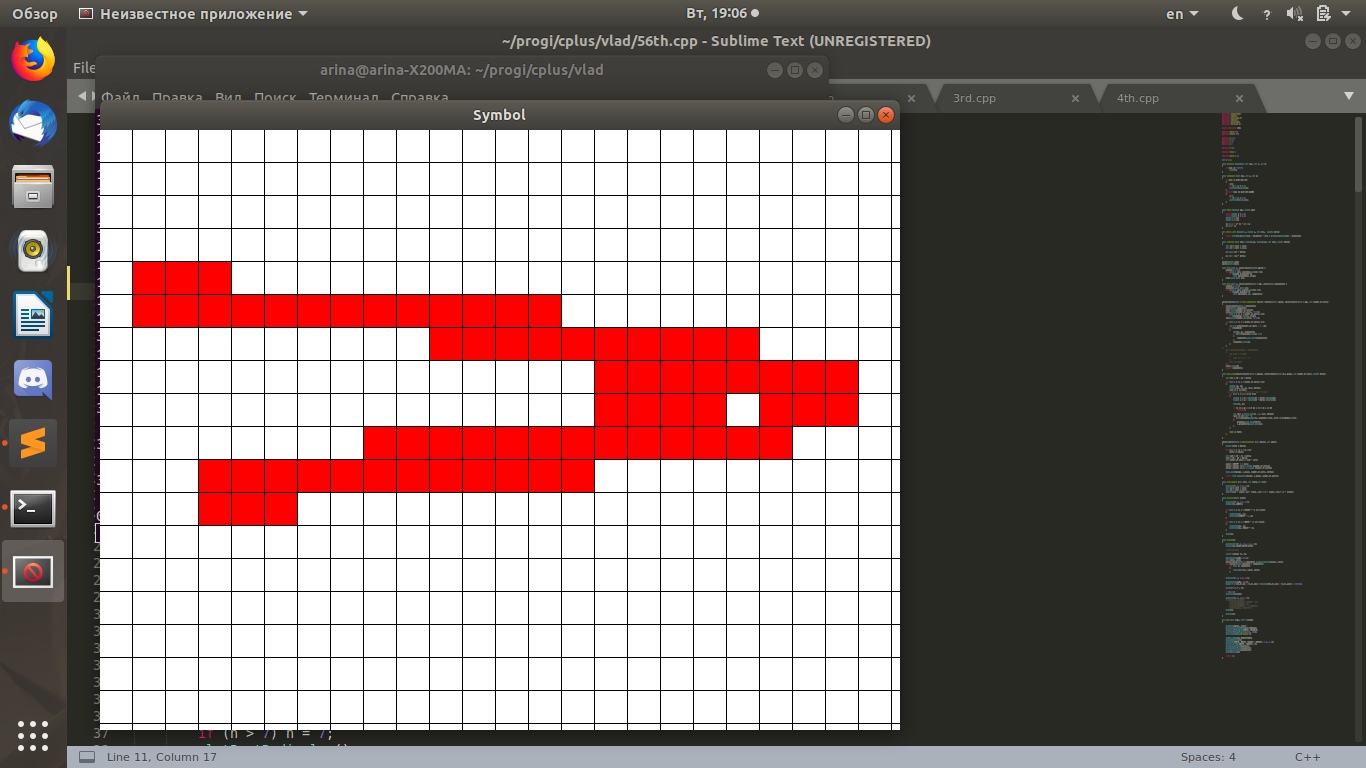
**Алгоритм локализации цепно-рекуррентного множества состоит в следующем:**

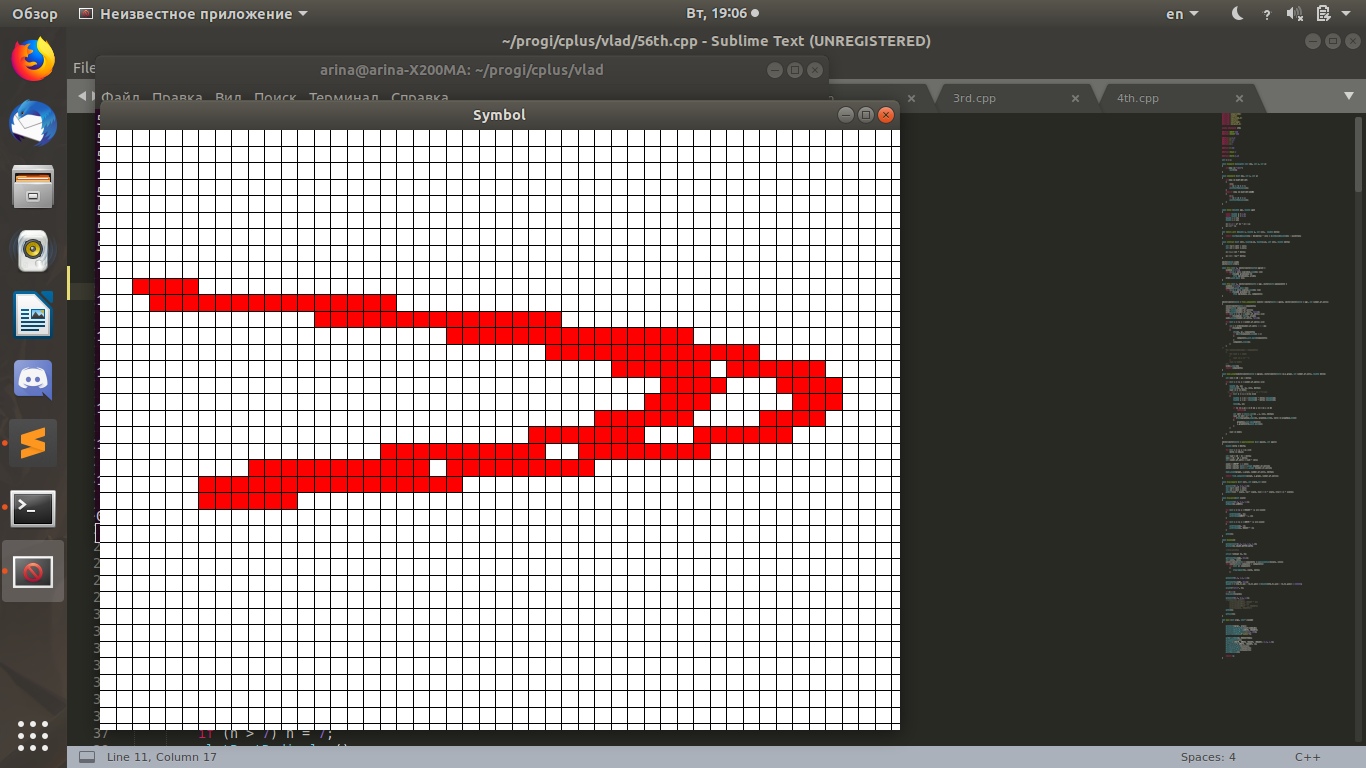
1. Строим исходное покрытие C компакта D. Находим символический образ G отображения f. Заметим, что ячейки исходного покрытия могут иметь произвольный диаметр d0.
2. Выделяем на графе G возвратные вершины {𝑖𝑘}. Если множество таких вершин пустое, значит локализуемое цепно-рекуррентное множество является пустым и процесс его локализации прекращается. Иначе,
3. используя их, находим замкнутую окрестность P = {x ∈ 𝐷𝑖𝑘 : 𝑖𝑘 — возвратные} цепно-рекуррентного множества Q.
4. Разбиваем ячейки {𝐷𝑖𝑘}, соответствующие возвратным вершинам символического образа и, таким образом, определяем новое покрытие.
5. Строим символический образ G для нового покрытия.
6. Переходим ко второму пункту, если размеры ячеек построенного символического образа достаточно велики. Повторяя процесс последовательного измельчения покрытия, мы получаем последовательность окрестностей 𝑃1,𝑃2,𝑃3. . . цепно-рекуррентного множества 𝑄 и последовательность наибольших диаметров 𝑑1,𝑑2,𝑑3, . . . ячеек, соответствующих возвратным вершинам символического образа для покрытия 𝐶𝑘

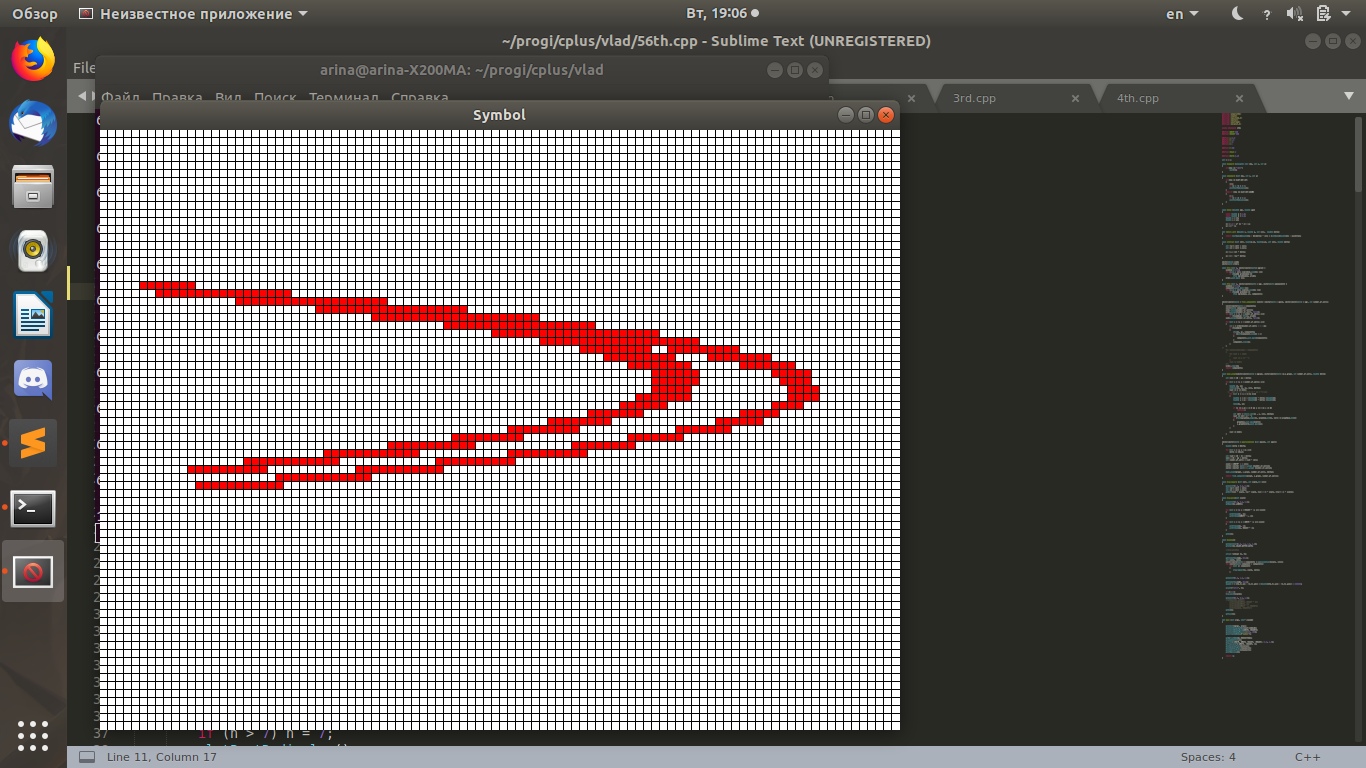
Для данной программы использовался язык программирования С++ на операционной системе Linux, дистрибутив Ubuntu 18.04.2 LTS Bionic Beaver.

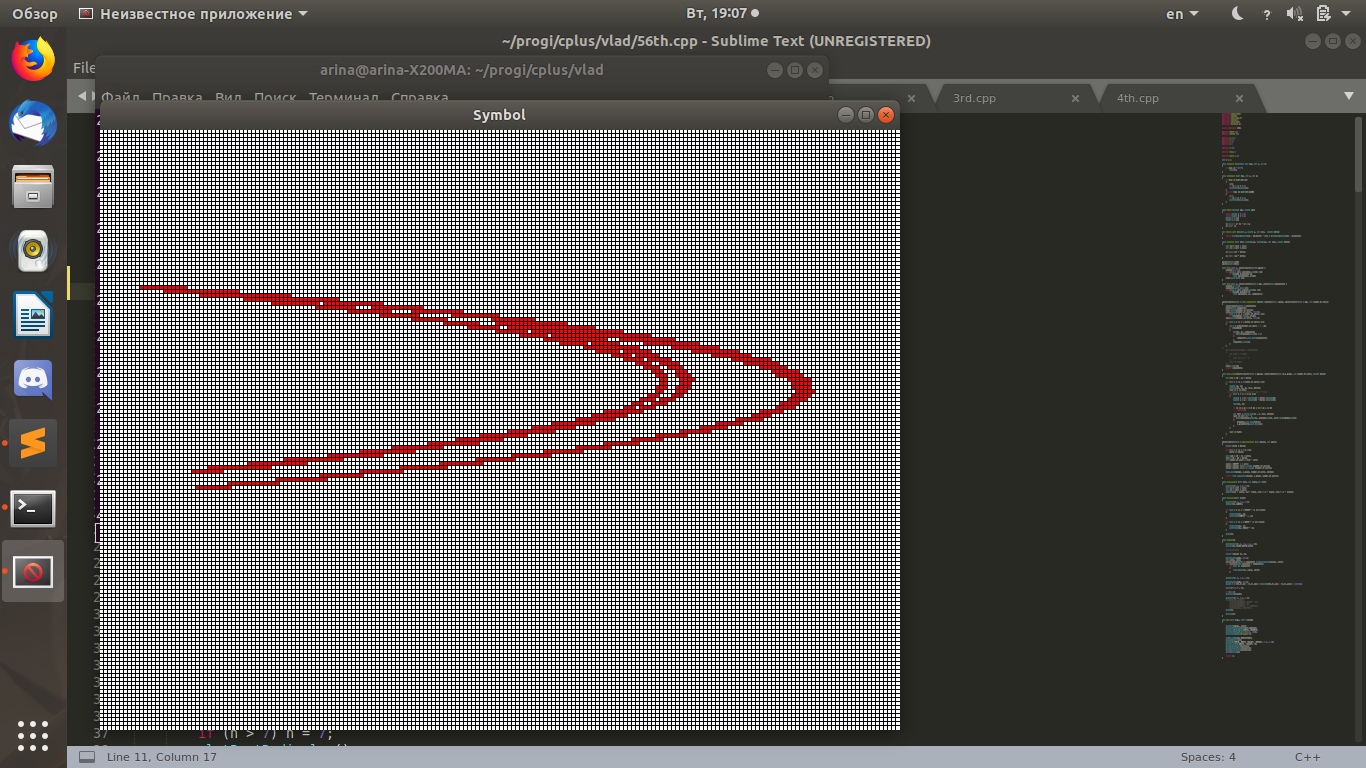
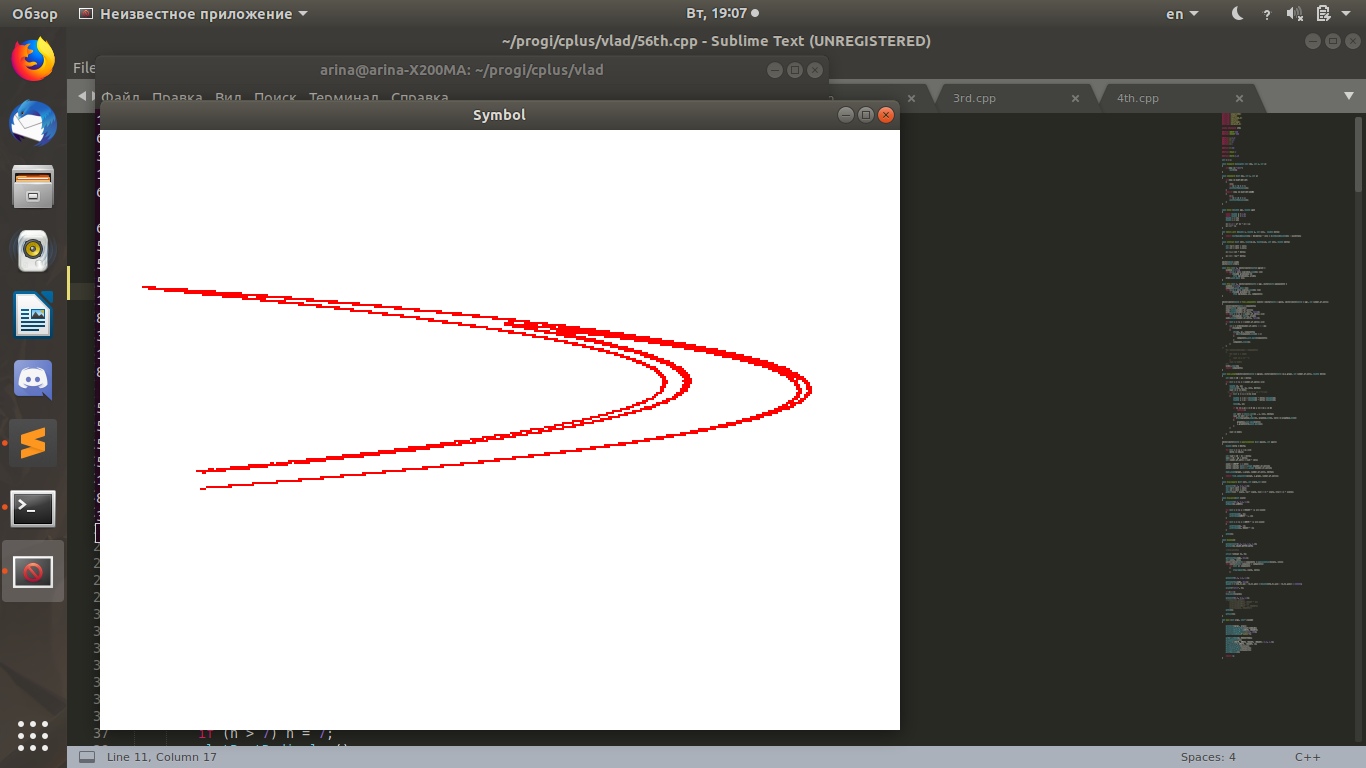
Ниже приведены результаты для 7 последовательных итераций.

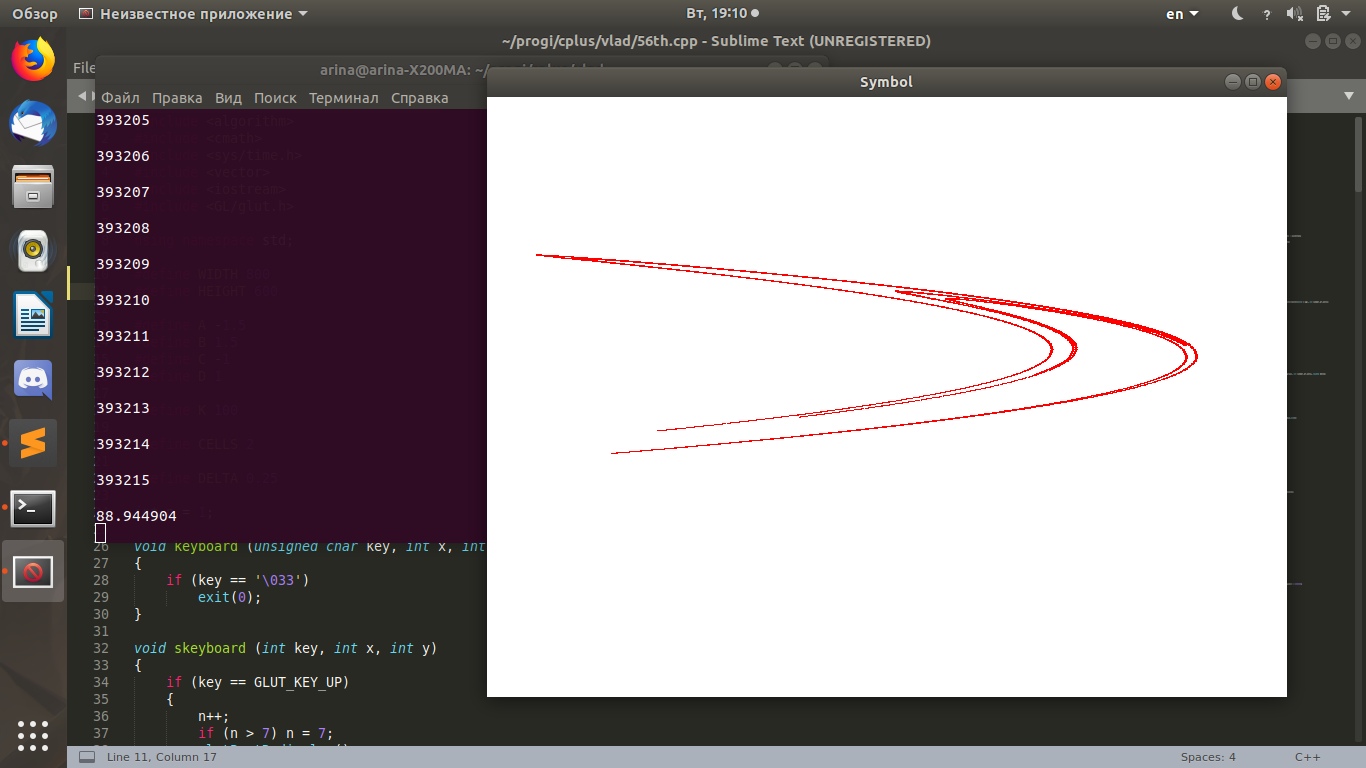










Время выполнения по итерациям:

1. 0.006573 с
2. 0.01517 с
3. 0.055561 с
4. 0.0739597 с
5. 2.953198 с
6. 11. 757658 с
7. 11. 799606 с