**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

[Институт прикладных информационных технологий](http://rasp.sstu.ru/) и коммуникаций

Кафедра Информационная безопасность автоматизированных систем

Направление подготовки 10.03.01 Информационная безопасность

**Расчётно-графическая работа**

по дисциплине «Языки программирования»

**«Определение маршрута с учетом загрузки сети»**

|  |  |
| --- | --- |
|  | Выполнил: студент 1 курса  учебной группы б-ИФБС11  очной формы обучения  Гришова Дарья Витальевна  Проверил:  ассистент каф. ИБС  Романчук С. П. |

Саратов 2020

Оглавление

[ЗАДАНИЕ 3](#__RefHeading__172_1926437900)

[ВВЕДЕНИЕ 4](#__RefHeading__176_1926437900)

[ОПИСАНИЕ ЗАДАЧИ 5](#__RefHeading__178_1926437900)

ОПИСАНИЕ РАБОТЫ…….…………………………………..………6

ОПИСАНИЕ ПРОГРАММЫ………….………………………………7

ЗАКЛЮЧЕНИЕ………………………………………………………10

ИСТОЧНИКИ……..…………………………………………..............11

# **Задание**

Дан файл с описанием карты города. Каждая связь имеет два веса – максимальная

пропускная способность дороги и ее текущий уровень загрузки в процентах. Возможный вариант структуры файла приведен ниже (комментарии в реальном файле будут

отсутствовать):

N – число вершин.

X1 Y1 // координаты вершины

X2 Y2

...

XN YN

M – число связей.

N1 K1 P1 V1 // какие вершины связаны, пропускная способность и

загрузка

N2 K2 P2 V2

...

NM KM PM VM

С учетом пробок проложить наиболее быстрый маршрут из пункта A в пункт B.

# **ВВЕДЕНИЕ**

**Поиск маршрута** — термин в информатике и искусственном интеллекте, который означает определение компьютерной программой наилучшего, оптимального маршрута между двумя точками.

# 

# **ОПИСАНИЕ ЗАДАЧИ**

По своей сути алгоритм поиска пути ищет на графе, начиная с одной (стартовой) точки и исследуя смежные узлы до тех пор, пока не будет достигнут узел назначения (конечный узел). Кроме того, в алгоритмах поиска пути в большинстве случаев заложена также цель найти самый короткий путь.

Ход работы:

* Создать возможность выбора количества дорог.
* Создать возможность выбора начальной точки.
* Создать возможность выбора конечной точки.
* Разработать возможность чтения карты из файла.
* Реализовать алгоритм поиска наикротчайшего пути между точками.
* После проделанной работы протестировать приложение самостоятельно.
* По итогу тестирования исключить все возможные неисправности.

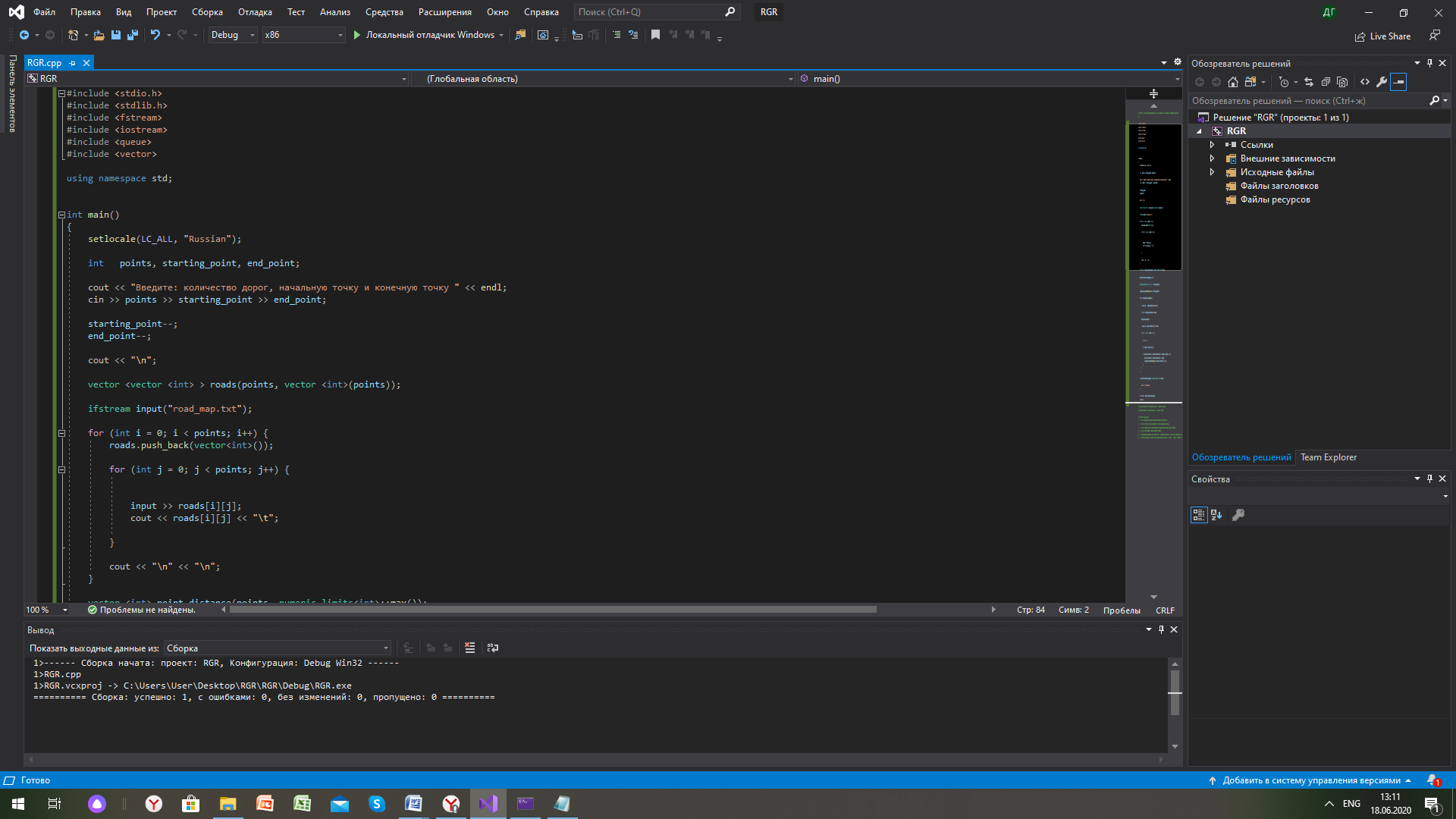
# **ОПИСАНИЕ РАБОТЫ**

Для разработки была выбран стандартный шаблон std::vector<T> — реализация динамического массива. Шаблон vector расположен в заголовочном файле <vector>. Как и все стандартные компоненты, он расположен в пространстве имён std. Данный интерфейс эмулирует работу стандартного массива C (например, быстрый произвольный доступ к элементам), а также некоторые дополнительные возможности, вроде автоматического изменения размера вектора при вставке или удалении элементов.

Программа должна быть написана на языке программирования C++. Разработка происходит в интегрированной среде разработки программного обеспечения и ряда других инструментальных средств – Microsoft Visual Studio на языке программирования C++.

**ОПИСАНИЕ ПРОГРАММЫ**

В начале работы программы мы задаём количество дорог, начальную точку и конечную точку. После чего приложение считывает карту дорог из файла и ищет наикротчайший путь до точки. Поиск пути осуществляется по алгоритму Дейкстры с помощью матрицы смежности. Алгоритм работает пошагово — на каждом шаге он «посещает» одну вершину и пытается уменьшать метки. Начальная точка полагается равной 0, метки остальных вершин — бесконечности. Работа алгоритма завершается, когда все вершины посещены.



# 

# 

# **ЗАКЛЮЧЕНИЕ**

Поставленная задача была выполнена, особых недочётов не было обнаружено.

Однако, есть несколько функций, которые можно было бы добавить. Например, поиск по нескольким картам, демонстрация пути, что своим образом разнообразило бы проект. Тем не менее версия приложения работает правильно и стабильно, а значит первоначальная задача была успешно выполнена.

# **ИСТОЧНИКИ**

1. https://ru.wikipedia.org/wiki/Vector\_(C%2B%2B)
2. https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B
3. https://code-live.ru/post/cpp-vector/
4. <https://ravesli.com/urok-95-vvedenie-v-std-vector-vektory/>
5. <https://ru.stackoverflow.com/questions/57957/%D0%94%D0%B2%D1%83%D0%BC%D0%B5%D1%80%D0%BD%D1%8B%D0%B9-vector>
6. https://qna.habr.com/q/218367