Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Лабораторная работа № 5**

Программирование на С

по дисциплине «Низкоуровневое программирование»

Выполнил

студент гр. 3530901/90004

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Остапчук А.С.

(подпись)

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексюк А.О.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Санкт-Петербург   
2021

**Задача**

1. Разработать статическую библиотеку, реализующую определенный вариантом задания абстрактный тип данных.

2. Разработать демонстрационную программу – консольное приложение, обеспечивающее ввод данных из файла (файлов), их обработку и вывод в файл (файлы); имена файлов передаются в качестве параметров командной строки.

**Вариант задания**

Вариант: 6 - Ненаправленный граф, представление матрицей смежности.

**Выполнение работы**

Напишем программу, реализующую ненаправленный граф, представление матрицей смежности. Так же разработаем демонстрационную программу и модульные тесты. Код находится: <https://github.com/Br0mm/LowLevelProgramming2021>

Реализация графа представляет из себя динамическую структуру данных. Она основана на структуре graphStruct, которая состоит из:

struct graphStruct {  
 int\*\* matrix; //матрица смежности размером size \* size, меняется динамически  
 int size; //размер нашей матрицы == количеству вершин  
};

typedef struct graphStruct \*graph;

**Основные функции библиотеки**

graph createGraph(int size); //создание графа с количеством вершин и размером size

void addEdge(int vertex1, int vertex2, graph currentGraph); //добавление ребра между 2 вершинами

void removeEdge(int vertex1, int vertex2, graph currentGraph); //удаление ребра между 2 вершинами

void addVertex(graph currentGraph); //добавление вершины в граф

void removeVertex(graph currentGraph, int vertex); //удаление вершины из графа

void printGraphFile(graph currentGraph, FILE \*outFile); //печать содержимого графа в файл

int getSize(graph currentGraph); //получение размера(количества вершин) графа

int\*\* getMatrix(graph currentGraph); //получение матрицы смежности графа

void deinitializeGraph(graph currentGraph); //освобождение памяти

**Описание программы примера**

Демонстрационная программа была написана в main.c. Данная программа создает граф заданного из консоли размера, добавляет рёбра между вершинами и выводит граф в выходной файл.

Входные параметры программы: [n] [k] [<edges\_file>] [<output\_file>]

* Параметр n задаёт размер создаваемого графа
* Параметр k задаёт количество переходов между вершинами из файла [<edges\_file>]
* Параметр [<edges\_file>] указывает входной файл
* Параметр [<output\_file>] указывает выходной файл

Для сборки библиотеки достаточно выполнить команду make в корневой папке клонированного проекта.

Пример запуска программы из консоли:



Пример запуска из Clion:



**Вывод**

В ходе выполнения лабораторной работы была разработана на языке С статически линкуемая библиотека, реализующая ненаправленный граф, представленный матрицей смежности. Так же были созданы модульные тесты и демонстрационная программа.