



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт радиоэлектроники и информатики
Кафедра геоинформационных систем

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 12
«Элементы алгоритмизации и процедурного
программирования»
по дисциплине
«ИНФОРМАТИКА»

Выполнил студент группы *ИКБО-51-23*

Лазаренко С. А.

Принял
Ассистент

Корчемная А.И.

Практическая
работа выполнена

«10» декабря 2023 г.

«Зачтено»

« » _____ 2023 г.

Москва 2023

СОДЕРЖАНИЕ

1	ПОСТАНОВКА ЗАДАЧИ	3
2	ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ	4
2.1	Блок-схемы алгоритмов программы	4
2.2	Код программы с комментариями	9
2.3	Проверка на работоспособность	13
2.4	Проверка на работоспособность при вводе некорректных значений	14
3	ВЫВОДЫ	15
4	СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	16

1 ПОСТАНОВКА ЗАДАЧИ

Требуется разработать блок-схему алгоритма и написать программу обработки данных в соответствии с выбранным и согласованным с преподавателем вариантом (см. далее). При этом требуется контролировать типы и диапазоны вводимых данных, а также предусмотреть обработку других исключительных ситуаций (если они есть), например, ситуацию деления на ноль. Блок-схема должна быть полной, т.е. должна описывать и процесс диалога с пользователем, и контроль вводимых данных, и подпрограммы вычислений с обработкой возможных исключительных операций. Блок-схема должна изображаться по ГОСТу. При обнаружении ошибки ввода или ошибки вычислений программа должна информативно уведомлять пользователя о причине ошибки. Если ошибка произошла на этапе ввода данных, то программа должна просить пользователя повторить ввод.

Дано: Создать квадратную матрицу размера $M \times M$, где M является целым числом из диапазона $[2, 5]$. Конкретный размер матрицы задается пользователем. Матрица должна содержать слова из 4х букв английского алфавита, которые могут быть как случайными, так и вводиться пользователем. На основе матрицы сформировать одномерный массив, куда должны войти слова, не содержащие определенную букву, которую задает пользователь. Полученный массив отсортировать по убыванию количества согласных букв в словах. Результаты обработки матрицы вывести на экран.

2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

2.1 Блок-схемы алгоритмов программы

На основе данного задания была построена блок-схема, на которой описывается программа-решение этого задания (рис. 1 – рис. 5). За основу были взяты основные действия с матрицами, а процесс диалога с пользователем был местами вынесен в функции.

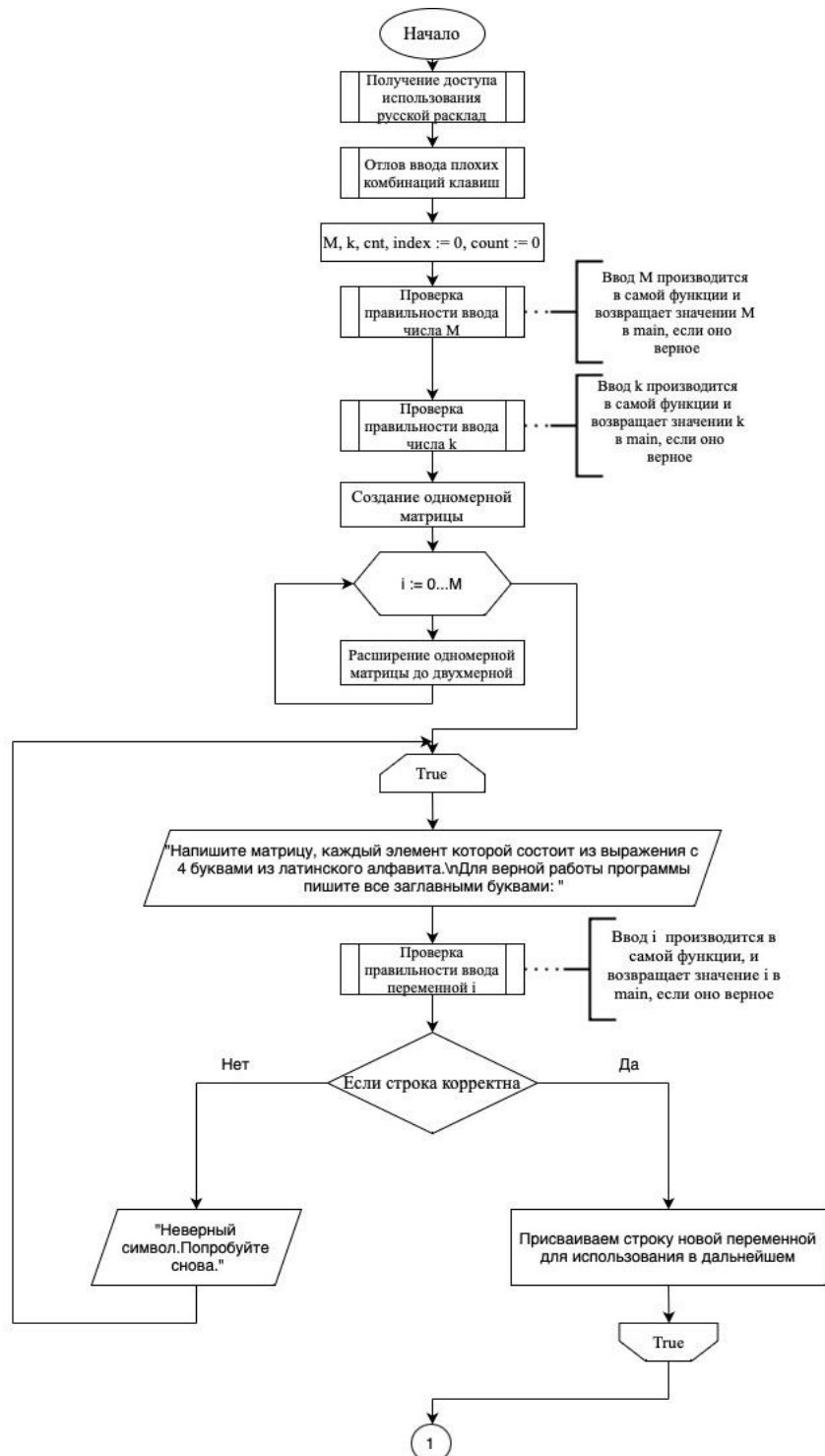


Рисунок 1 – реализация функции Main (часть 1)

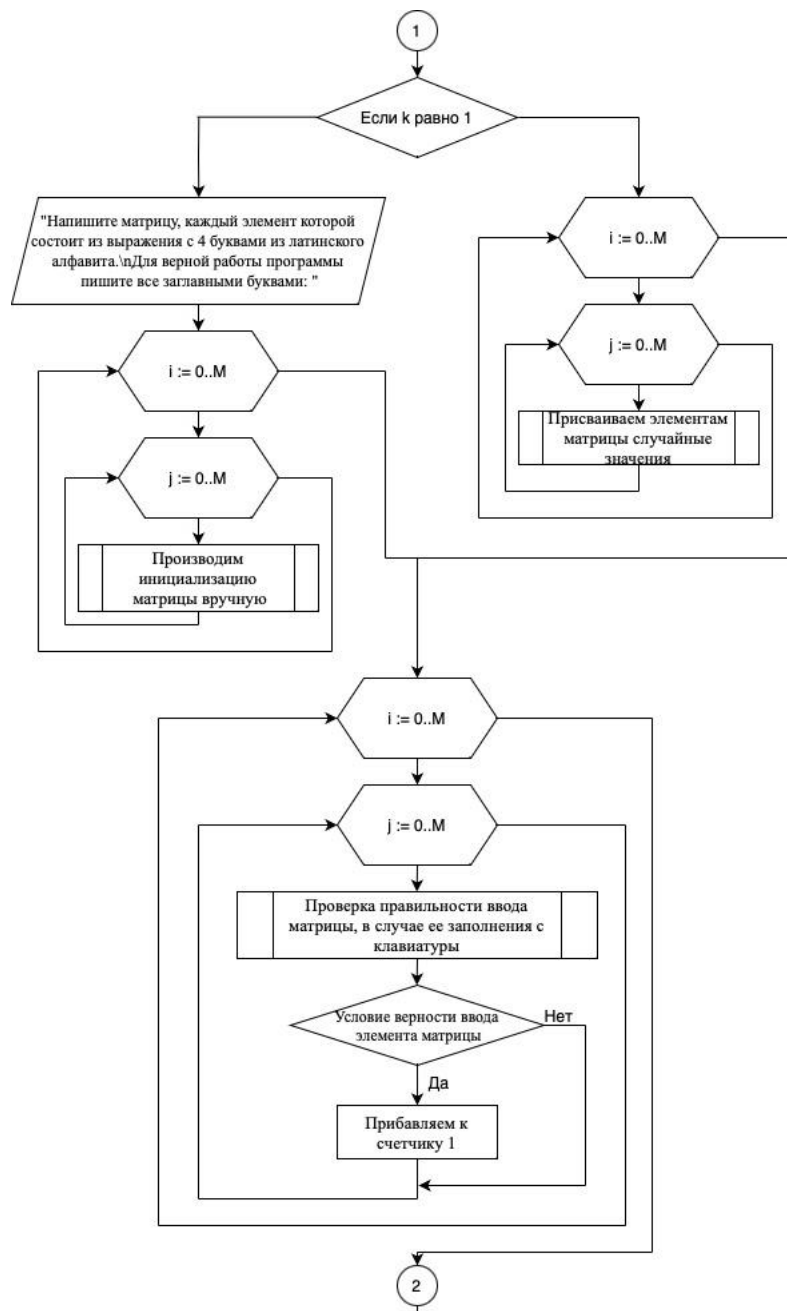


Рисунок 2 – реализация функции Main (часть 2)

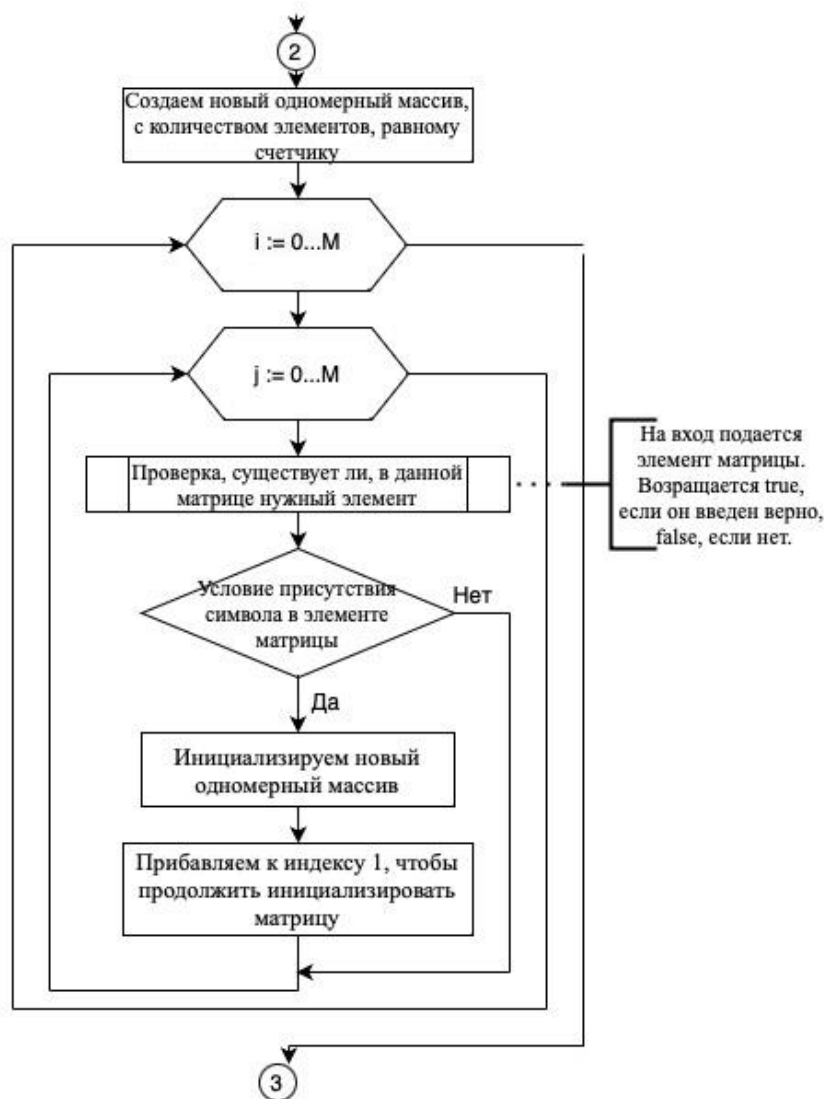


Рисунок 3 – реализация функции Main (часть 3)

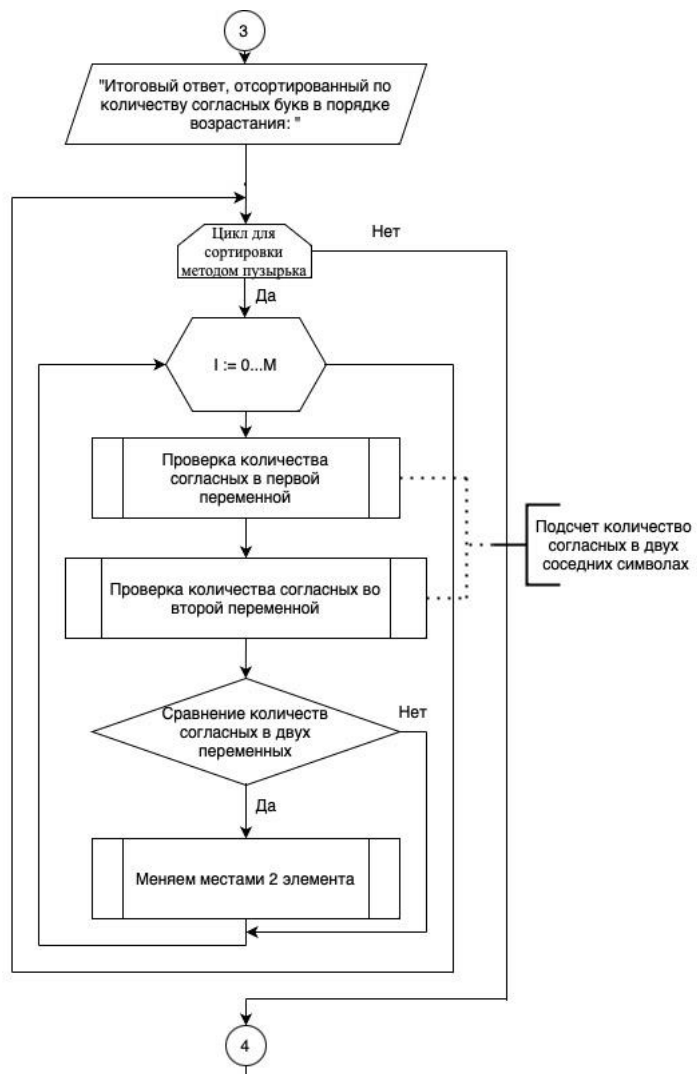


Рисунок 4 – реализация функции Main (часть 4)



Рисунок 5 – реализация функции Main (часть 5)

2.2 Код программы с комментариями

На рисунках 6,7,8,9,10,11 представлен полный код программы, написанной на языке C++ в среде Xcode.

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string>
#include <sstream>
#include <stdio.h>
#include <signal.h>
#include <stdlib.h>
#include <cstdlib>

using namespace std;

void my_handler(sig_atomic_t s) { // Извлечение числа из потока ввода
    cout << " Повторите." << std::endl;
    exit(1);
}

bool razmer_matrix(int M) // Проверка на корректный размер матрицы
{
    if (M >= 2 && M <= 5) { // условие соответствие числа в нужном диапазоне
        return true;
    }
    return false;
}

int check() // проверка на корректный ввод числа
{
    int number; // инициализация переменной
    std::string input;
    while (true) {
        cout << "Напишите M, где M: [2;5] и целое значение: ";
        std::getline(std::cin, input);
        if (cin.eof()) { // Проверка на наличие пробелов
            cout << "Введена запрещенная комбинация клавиш для данной программы." << std::endl;
            exit(1);
        }
        if (input.find(' ') != std::string::npos) { // Проверка на правильность ввода числа
            cout << "Введено неверное число! Напишите еще раз." << std::endl;
            continue;
        }
        std::istringstream iss(input);
        if (!(iss >> number)) { // проверка на извлечение символа, являющегося числом
            cout << "Введено неверное число! Напишите еще раз." << std::endl;
            continue;
        }
        if (iss.rdbuf()->in_avail() != 0) { // проверка на отсутствие символов после введенного числа
```

Рисунок 6 — Реализация кода с комментариями на C++ в среде Xcode (часть 1)

```

        cout << "Введено неверное число! Напишите еще раз." << std::endl;
        continue;
    }
    if (iss.rdbuf()->in_avail() != 0) { // проверка на отсутствие символов после введенного числа
        cout << "Введено неверное число! Напишите еще раз." << std::endl;
        continue;
    }
    if (razmer_matrix(number) == false) { // проверка на верно введенное значение, соответствующее диапазону
        cout << "Введено неверное число! Напишите еще раз." << std::endl;
        continue;
    }
    cout << "M = " << number << std::endl;
    break;
}
return number;
}

int check_p()
{
    int number; // инициализация переменной
    std::string input;

    while (true) {
        cout << "Напишите, как вы хотите инициализировать матрицу.\n 1 -> ввести значения матрицы с клавиатуры.\n 2 -> заполнить матрицу случайным
            std::endl;
        std::getline(std::cin, input);
        if (cin.eof()) { // проверка на обнаружение запрещенной комбинации клавиш
            cout << "Введена запрещенная комбинация клавиш. Перезапустите программу!" << std::endl;
            exit(1);
        }
        if (input.find(' ') != std::string::npos) { // проверка на наличие пробелов
            cout << "Введено неверное число! Напишите еще раз." << std::endl;
            continue;
        }

        std::istringstream iss(input);
        if (!(iss >> number)) { // проверка на успешное извлечение из потока
            cout << "Введено неверное число! Напишите еще раз." << std::endl;
            continue;
        }
        if (iss.rdbuf()->in_avail() != 0) { // проверка на наличие дополнительных символов
            cout << "Введено неверное число! Напишите еще раз." << std::endl;
            continue;
        }
        if (number != 1 && number != 2) { // проверка на корректность введенного числа
            cout << "Введено неверное число! Напишите еще раз." << std::endl;

```

Рисунок 7 — Реализация кода с комментариями на C++ в среде Xcode (часть 2)

```

        cout << "Введено неверное число! Напишите еще раз." << std::endl;
        continue;
    }
    cout << "Введенное число: " << number;
    if (number == 2) { cout << ", заполним матрицу случайными значениями.\n"; }
    else if (number == 1) { cout << ", матрица заполняется с клавиатуры.\n"; }
    break;
}
return number;
}

string getRandomLetter(int low = 65, int high = 90) // Функция для случайного заполнения матрицы
{
    string k; int cnt = 0; // инициализация переменных
    while (cnt != 4) {
        k += (char)(rand() % (high - low + 1) + low); // случайная генерация строк
        cnt += 1;
    }
    return k;
}

string first_check() //
{
    std::string input;
    bool hasSpace = true;

    while (hasSpace) {
        std::getline(std::cin, input);
        if (cin.eof()) { // проверка на обнаружение запрещенной комбинации клавиш
            cout << "Введена запрещенная комбинация клавиш. Перезапустите программу!" << std::endl;
            exit(1);
        }

        if (input.find(' ') != std::string::npos) { // проверка на наличие пробелов при вводе
            cout << "Неверный символ. Напишите еще раз" << std::endl;
        }
        else {
            hasSpace = false;
        }
    }
    return input;
}

bool second_check(string l)
{
    string str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

```

Рисунок 8 — Реализация кода с комментариями на C++ в среде Xcode (часть 3)

```

    for (int i = 0; i < str.length(); ++i) { // проверка на содержание правильных букв
        if (l[0] == str[i]) return true;
    }
    return false;
}

int countConsonants(std::string str) { // Функции для подсчета согласных
    int count = 0;
    for (int i = 0; i < str.length(); ++i) { // Условие для верного подсчета согласных
        if (str[i] == 'B' || str[i] == 'C' || str[i] == 'D' || str[i] == 'F' || str[i] == 'G' || str[i] == 'H' || str[i] == 'J' || str[i] == 'K' || str[i] == 'L' ||
            str[i] == 'M' || str[i] == 'N' || str[i] == 'P' || str[i] == 'Q' || str[i] == 'R' || str[i] == 'S' || str[i] == 'T' || str[i] == 'V' || str[i] == 'W' ||
            str[i] == 'X' || str[i] == 'Z') {
            count++;
        }
    }
    return count;
}

string checkString() {
    string stringToCheck;
    while (true) {
        bool flag = true;
        std::getline(std::cin, stringToCheck);
        if (cin.eof()) { // проверка на обнаружение запрещенной комбинации клавиш
            cout << "Введена запрещенная комбинация клавиш. Перезапустите программу!" << std::endl;
            exit(1);
        }
        if (stringToCheck.find(' ') != std::string::npos) { // проверка на наличие пробелов при вводе
            cout << "Строка введена неверно. Повторите" << std::endl;
            continue;
        }
        if (stringToCheck.length() != 4) { // проверка на соответствие длины строки
            cout << "Строка введена неверно. Повторите." << std::endl;
            continue;
        }
        std::string characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        for (int i = 0; i < stringToCheck.length(); ++i) {
            if (characters.find(stringToCheck[i]) == std::string::npos) { // проверка, что все элементы строки являются заглавными латинскими буквами
                cout << "Неверно введена строка. Попробуйте снова." << std::endl;
                flag = false;
                break;
            }
        }
        if (flag == false) { continue; } // условие, что если строка не соответствует хотя бы одному из предыдущих условий функции, то она будет продолжать запрашивать
        // верный ввод
    }
    else {

```

Рисунок 9 — Реализация кода с комментариями на C++ в среде Xcode (часть 4)

```

        break;
    }
}
return stringToCheck;
}

bool SymbolString(string m, string str)
{
    if (m.find(str) != m.npos) { // проверка на содержание символа в строке
        return true;
    }
    return false;
}

int main()
{
    setlocale(LC_ALL, "");
    signal(SIGINT, my_handler);
    int M, k, cnt = 0, index = 0, count = 0; string str; // инициализация необходимых переменных
    M = check();
    k = check_p();
    string** matrix = new string * [M];
    for (int i = 0; i < M; ++i) {
        matrix[i] = new string[M];
    }
    while (true) {
        string l;
        cout << "Напишите символ, который должны содержать слова массива.\nДля верной работы программы вводите символ заглавными латинскими буквами: ";
        l = first_check(); // возвращение к данной функции в случае ввода l
        if (l.length() == 1 && second_check(l)) { str = l; break; } // проверка на корректность введенного числа
        else { cout << "Неверный символ. Попробуйте снова." << std::endl; }
    }
    if (k == 1) {
        cout << "Напишите матрицу, каждый элемент которой состоит из выражения с 4 буквами из латинского алфавита.\nДля верной работы программы пишите все заглавными
        буквами: " << std::endl;
        for (int i = 0; i < M; ++i) {
            for (int j = 0; j < M; ++j) {
                matrix[i][j] = checkString();
            }
        }
    }
    else {
        for (int i = 0; i < M; ++i) { // генерация случайной матрицы
            for (int j = 0; j < M; ++j) {
                matrix[i][j] = getRandomLetter();
            }
        }
    }
}

```

Рисунок 10 — Реализация кода с комментариями на C++ (часть 5)

```

        std::cout << matrix[i][j] << std::endl;
    }
}
}
for (int i = 0; i < M; ++i) {
    for (int j = 0; j < M; ++j) {
        if (SymbolString(matrix[i][j], str) == true) {
            cnt += 1;
        }
    }
}
string* new_matrix = new string[cnt]; // замена неотсортированной матрицы на отсортированную
for (int i = 0; i < M; ++i) {
    for (int j = 0; j < M; ++j) {
        if (SymbolString(matrix[i][j], str) == true) {
            new_matrix[index] = matrix[i][j];
            index += 1;
        }
    }
}
cout << "\nКонечный ответ, отсортированный по количеству согласных букв в порядке возрастания: " << std::endl; // вывод конечной матрицы
while (count <= pow(cnt, 2)) {
    for (int i = 0; i < cnt - 1; ++i) {
        if (countConsonants(new_matrix[i]) > countConsonants(new_matrix[i + 1])) { swap(new_matrix[i], new_matrix[i + 1]); }
    }
    ++count;
}
if (cnt == 0) { cout << "Строки с буквой " << str << " не обнаружено." << std::endl; } // проверка на наличие строк с нужной буквой
for (int i = 0; i < cnt; ++i) {
    std::cout << new_matrix[i] << std::endl;
}
return 0;
}

```

Рисунок 11 — Реализация кода с комментариями на C++ (часть б)

2.3 Проверка на работоспособность

После написания кода проверим правильность работы алгоритма и кода (Рис. 12,13).

```

Напишите M, где M: [2;5] и целое значение: 3
M = 3
Напишите, как вы хотите инициализировать матрицу.
1 -> ввести значения матрицы с клавиатуры.
2 -> заполнить матрицу случайными значениями.
2
Введенное число: 2, заполним матрицу случайными значениями.
Напишите символ, который должны содержать слова массива.
Для верной работы программы вводите символ заглавными латинскими буквами: F
LRFK
QYUQ
FJKX
YQVN
RTYS
FRZR

```

Рисунок 12 — Проверка работоспособности кода (часть 1)

```
FJKX
YQVN
RTYS
FRZR
MZLY
GFVE
ULQF

Конечный ответ, отсортированный по количеству согласных букв в порядке возрастания:
GFVE
ULQF
LRFK
FJKX
FRZR
Program ended with exit code: 0
```

Рисунок 13 — Проверка работоспособности кода (часть 2)

2.4 Проверка на работоспособность при вводе некорректных значений

```
Напишите М, где М: [2;5] и целое значение: /
Введено неверное число! Напишите еще раз.
Напишите М, где М: [2;5] и целое значение: 1
Введено неверное число! Напишите еще раз.
Напишите М, где М: [2;5] и целое значение: фу
Введено неверное число! Напишите еще раз.
Напишите М, где М: [2;5] и целое значение: 008шщйу
Введено неверное число! Напишите еще раз.
Напишите М, где М: [2;5] и целое значение: "
Введено неверное число! Напишите еще раз.
Напишите М, где М: [2;5] и целое значение: "
Введено неверное число! Напишите еще раз.
Напишите М, где М: [2;5] и целое значение: Введена запрещенная комбинация клавиш для данной программы.
Program ended with exit code: 1
```

Рисунок 14 — Проверка работоспособности при некорректных значениях

3 ВЫВОДЫ

Были получены навыки по созданию блок-схем. Улучшены навыки программирования. Тестирование показало работоспособность программы при корректных и некорректных вводах данных

4 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Смирнов, С. С. Методические указания по выполнению практических работ для студентов / С. С. Смирнов, Д. А. Карпов. – Москва : МИРЭА – Российский технологический университет, 2020. – 103 с.

2. Лекции по информатике / С.С. Смирнов — М., МИРЭА — Российский технологический университет.