|  |  |
| --- | --- |
| logo | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** | |
| Институт информационных технологий (ИТ) | |
| Кафедра Общей информатики | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ** | |
| **по дисциплине** | |
| **«**ИНФОРМАТИКА**»** | |
|  | |
| Выполнил студент группы \_\_\_\_ИВБО-03-18\_\_\_\_ | Кладов А.С. |
| Принял  Старший преподаватель | Смирнов С.С. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |

Москва 2018

Содержание

[Практическая работа №8 3](#_Toc531333310)

[Цель работы 3](#_Toc531333311)

[Таблица истинности 3](#_Toc531333312)

[Реализация на дешифраторе 4 – 16 и шифраторе 16 - 4 3](#_Toc531333313)

[Вывод 5](#_Toc531333314)

[Список информационных источников 6](#_Toc531333315)

# Лабораторная работа №4

# Цель работы

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.Таблица переходов для счетчика

Даны следующие исходные данные:

2.13. Создать квадратную матрицу размера MxM, 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, состоящий из количеств гласных букв в словах. Полученный массив отсортировать по возрастанию. Результаты обработки матрицы вывести на экран.

# Реализация программы на C++

Ниже представлен исходный код, реализующий данную программу:

#include <ctime>

#include <iostream>

#include <limits>

#include <string>

#include <vector>

using namespace std;

void checking(string &str, int size, int start, int end) {

cin >> str;

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Очистка входящего потока

for (int i = 0; i < size; i++) {//проверка всех букв на соответствие заданным критериям

while (

!((int)str[i] >= start && (int)str[i] <= end && str.length() == size)) {

cout << "Invalid input, try again" << endl;

cin >> str;

i = 0;

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

};

void sortingg(int a[], int start, int end){

int min = 0;

for(int i = start; i < end; i++){

min = i;

for(int k = i; k < end; k++){

if(a[k] < a[min]){min = k;}

}

int tmp = a[min];

a[min] = a[i];

a[i] = tmp;

}

}

int main() {

string str, str1;

int m = 0;

cout << "Enter value in range [2, 5]" << endl;

checking(str, 1, 50, 53);

cout << "If you wish to type in values manually, type 1, if not, type 2"

<< endl;

checking(str1, 1, 49, 50);

string a[5][5];//Объявление массива (большего размера он быть не может)

m = (int)str[0] - 48;//получение размера массива

if (str1 == "1") {

string str2;

for (int i = 0; i < m; i++) {

for (int k = 0; k < m; k++) {

checking(str2, 4, 65, 90);//проверка ввода

a[i][k] = str2;

}

}

for (int i = 0; i < m; i++) {//вывод массива

for (int k = 0; k < m; k++) {

cout<<a[i][k]<<' ';

}

cout<<endl;

}

} else {

srand(time(0));

for (int i = 0; i < m; i++) {

for (int k = 0; k < m; k++) {

for (int j = 0; j < 4; j++) {

a[i][k][j] = (char)(65 + rand() % 26);//Добавление случайного символа

cout << a[i][k][j];

}

cout << ' ';

}

cout << endl;

}

}

int mas[25]{0};//Из условий задачи массив не может быть больше 25

for (int i = 0; i < m; i++) {

for (int k = 0; k < m; k++) {

int result = 0;

for (int j = 0; j < 4; j++) {//поиск гласных

if (a[i][k][j] == 'A' || a[i][k][j] == 'E' || a[i][k][j] == 'I' ||

a[i][k][j] == 'O' || a[i][k][j] == 'U')

result++;

}

mas[i \* m + k] = result;//запись в массив

}

}

for (int i = 0; i < m \* m; i++) {//вывод массива

cout << mas[i] << ' ';

}

cout<<endl;

sortingg(mas, 0, m\*m);//сортировка

for (int i = 0; i < m \* m; i++) {//вывод отсортированного массива

cout << mas[i] << ' ';

}

return 0;

}

# Блок схема, реализующая алгоритм программы

Ниже представлены блок схемы, реализующий данную программу, при чем сначала идут две подпрограммы, которые позже будут вызваны в основной программе:

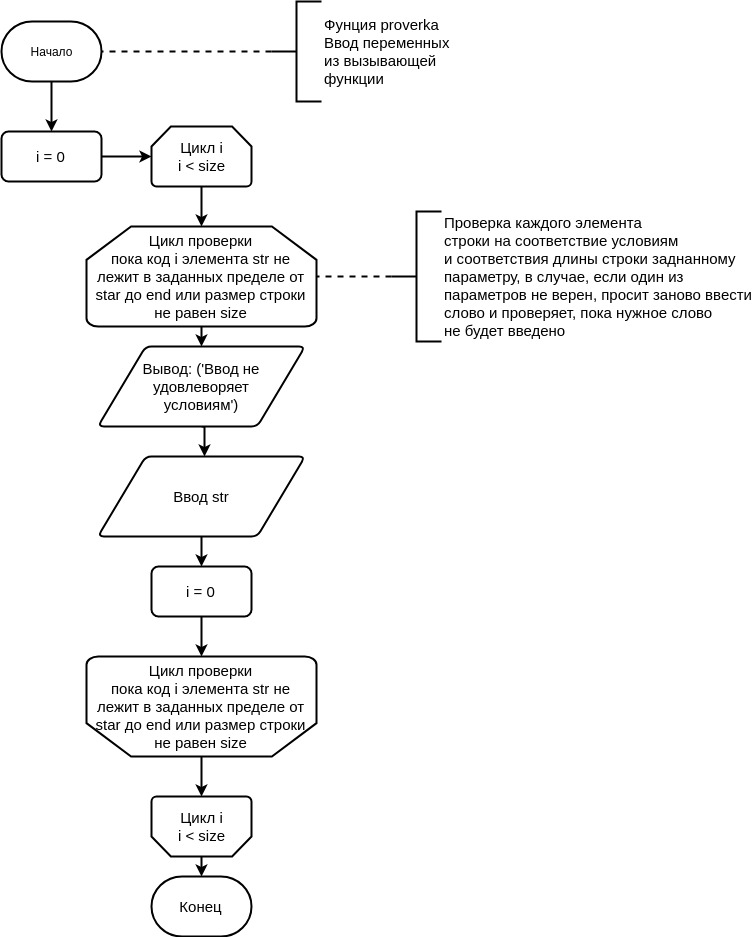


Рисунок 1. Подпрограмма proverka

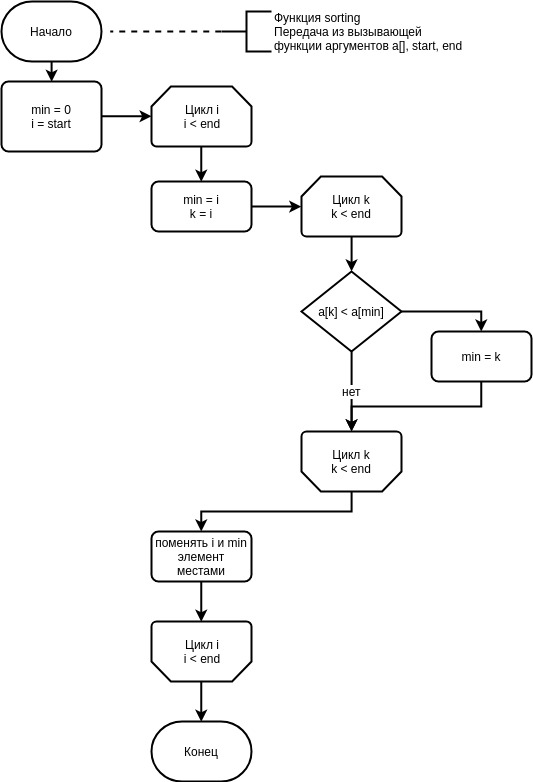


Рисунок 2. Подпрограмма sorting

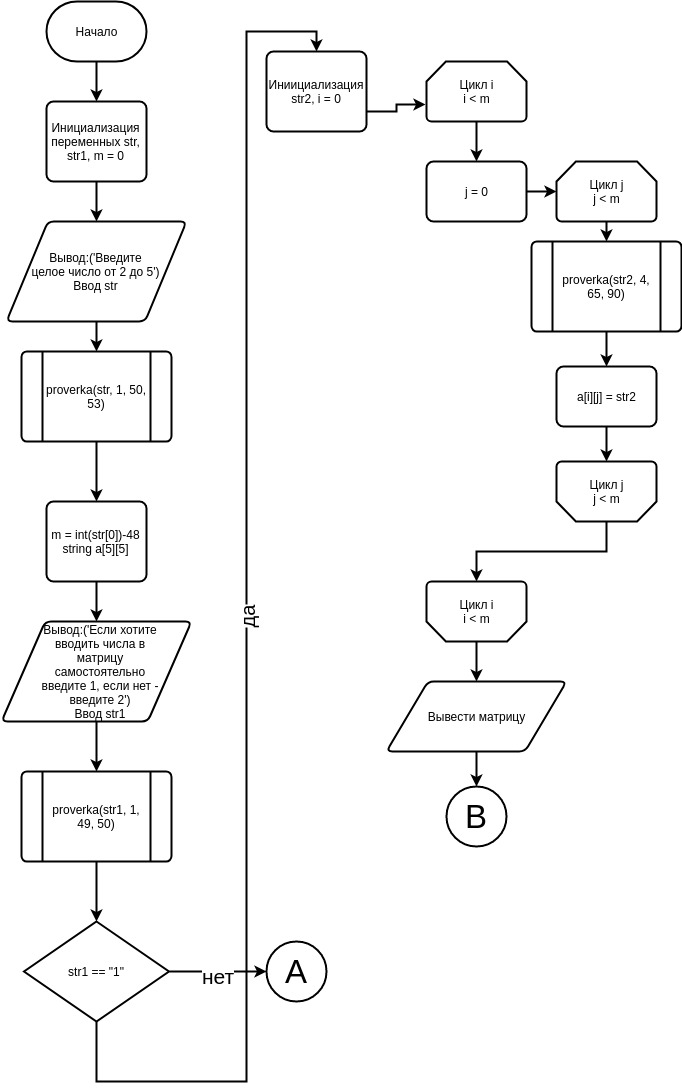


Рисунок 3. Основная программа 1/3

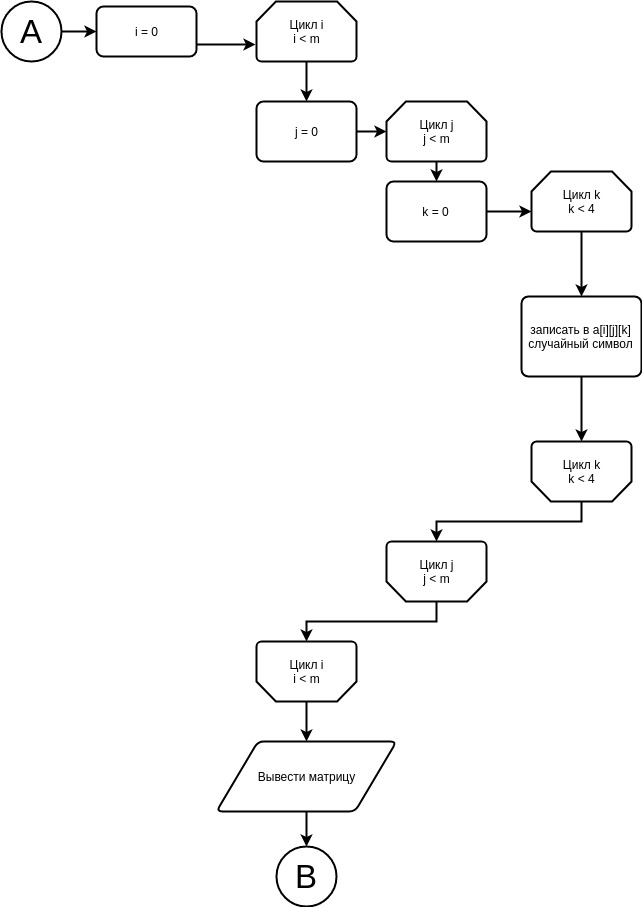


Рисунок 4. Основная программа 2/3

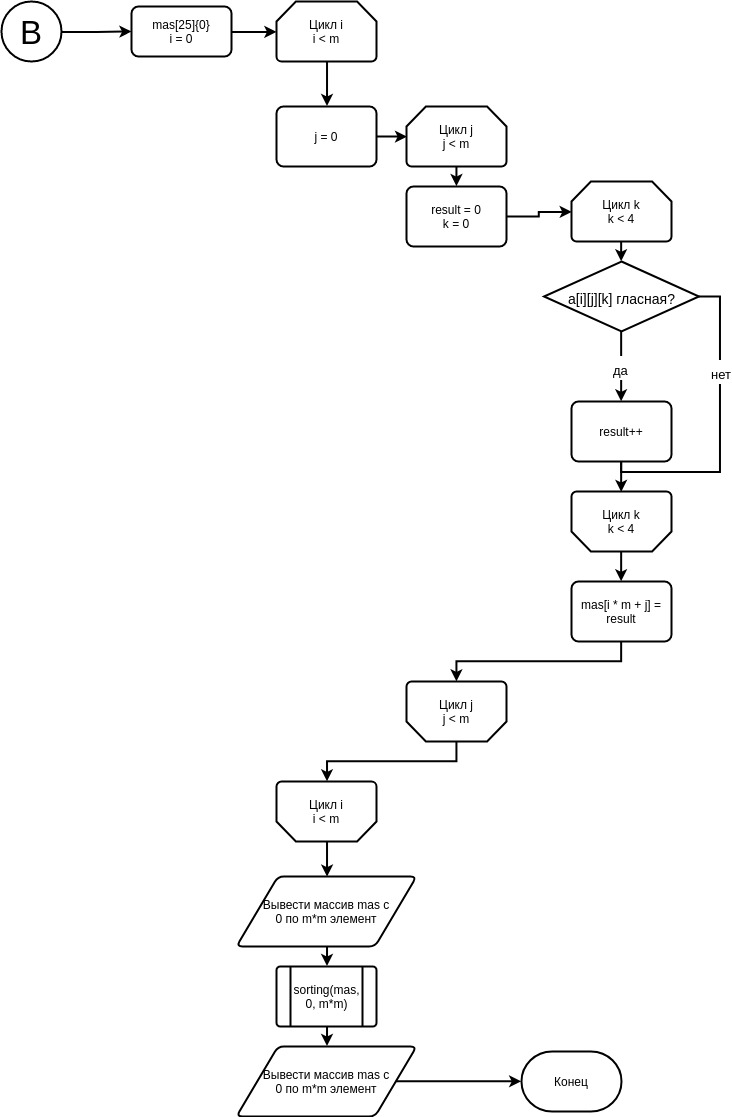


Рисунок 5. Основная программа 3/3

# Проверка работоспособности программы

Ниже представлены различные результаты работы программы в зависимости от вводимых значений:

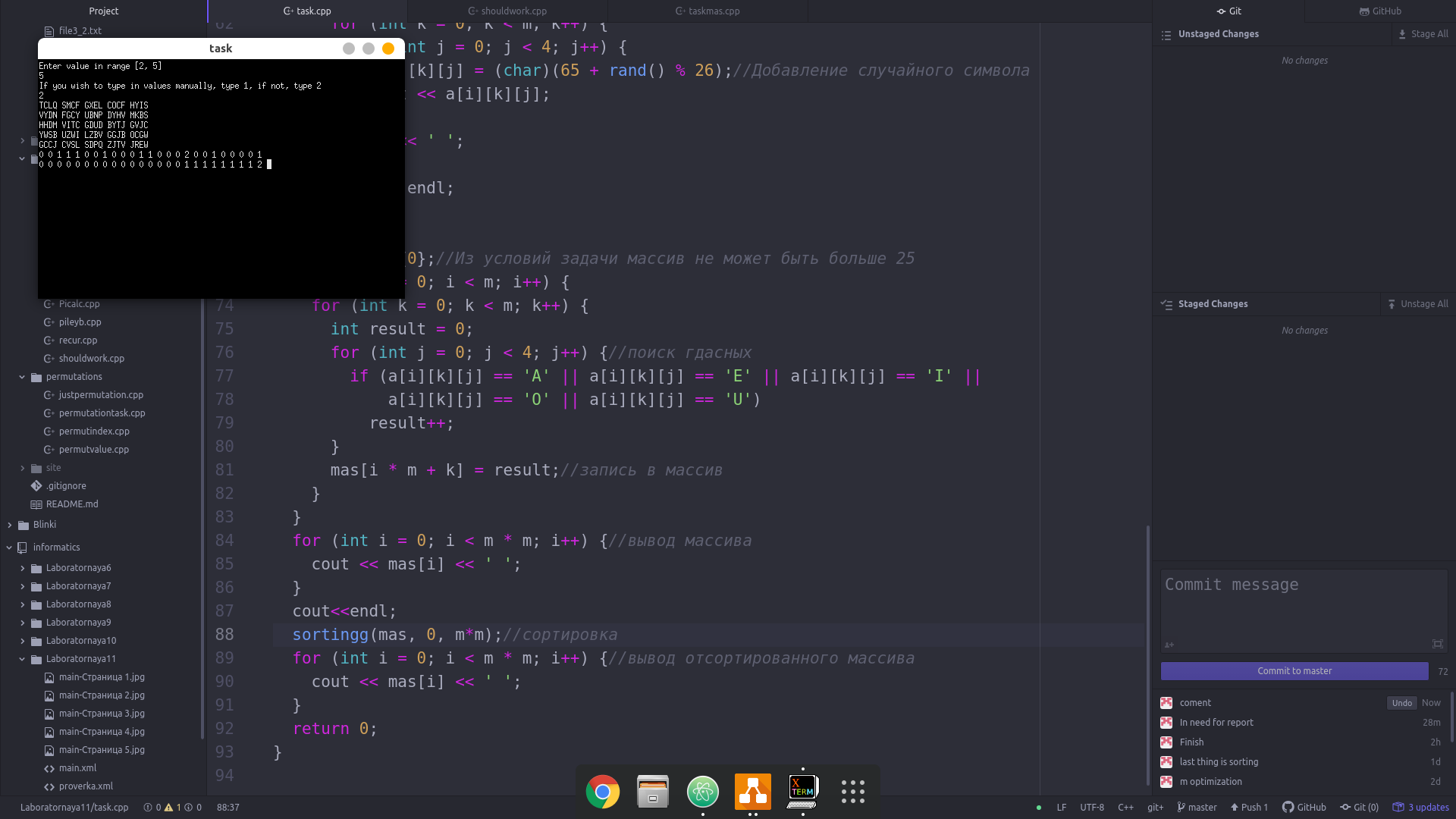


Рисунок 6. Вывод программы при вводе 5 и выборе случайного заполнения массива

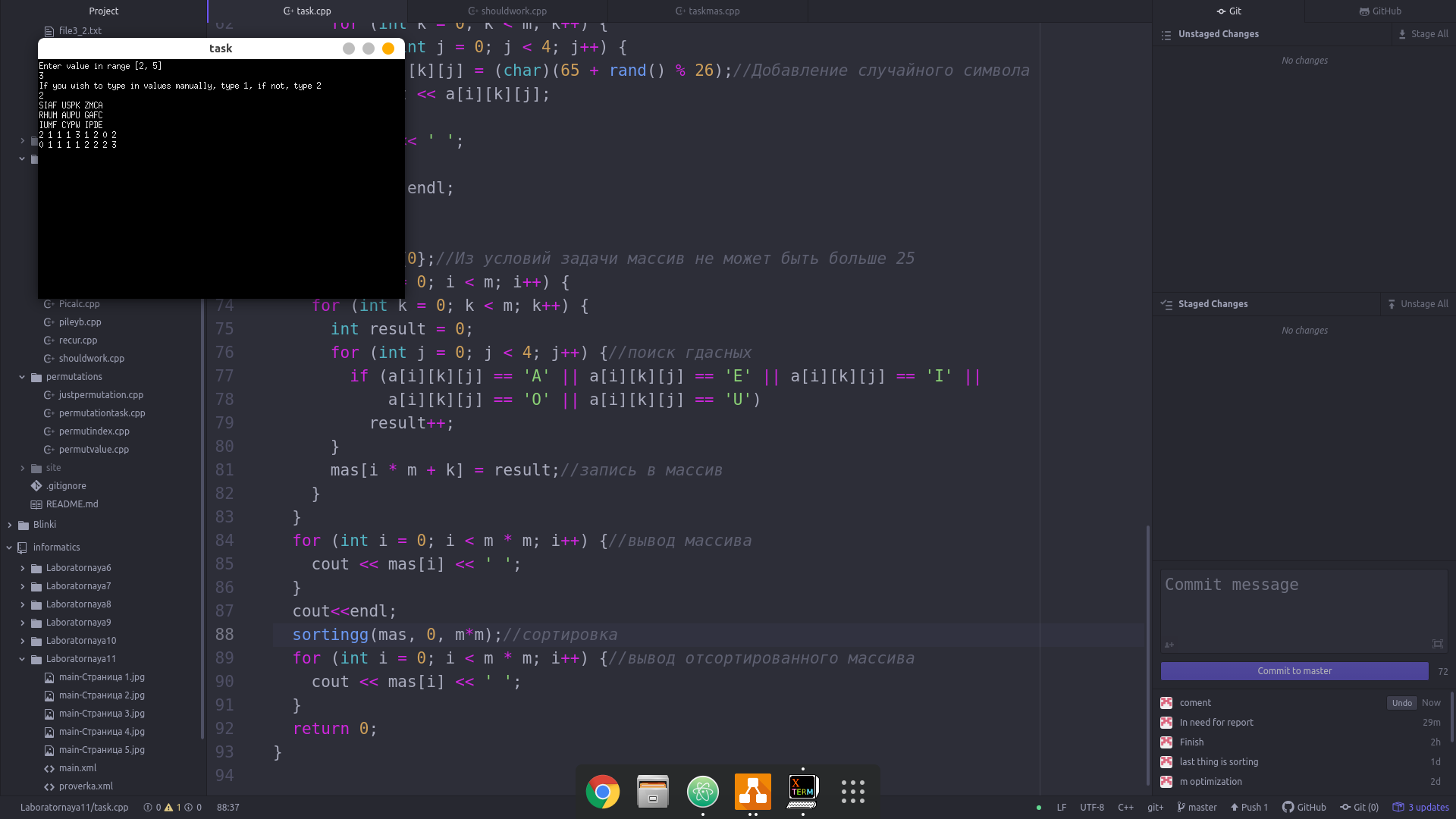


Рисунок 7. Вывод программы при вводе 3 и выборе случайного заполнения массива

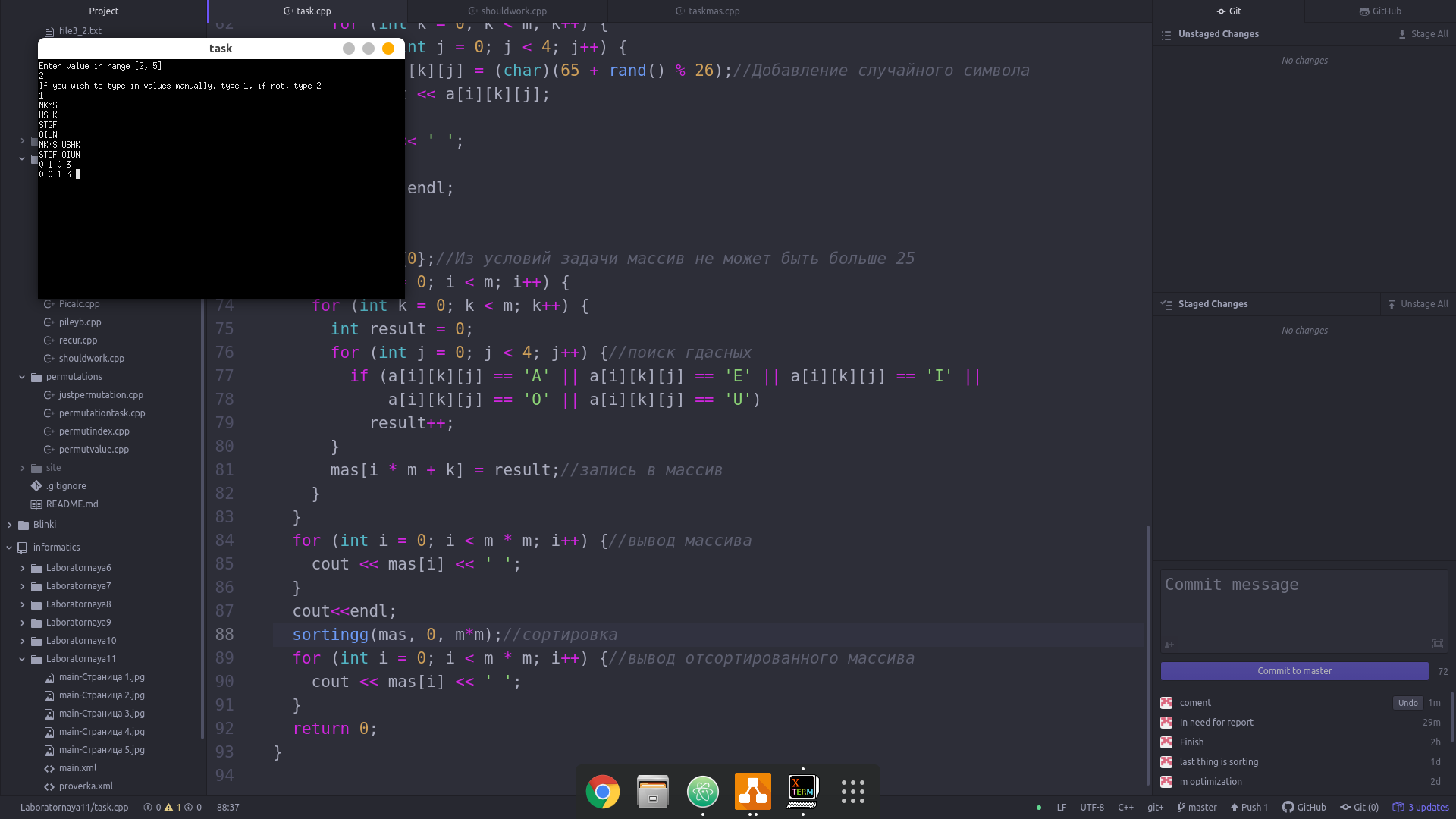


Рисунок 8. Вывод программы при вводе 2 и выборе ручного заполнения массива

Из вышеперечисленного убеждаемся в верности работы программы.

# Вывод

В ходе данной работы отработал навык составления и реализации алгоритмов, реализующих программу, и составление блок схем в соответствии с ГОСТом.

# Список информационных источников

1. Logisim официальная документация. [В Интернете] http://www.cburch.com/logisim/ru/docs.html.

2. Смирнов С.С. Информатика: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов—М., МИРЭА Российский технологический университет, 2018. –104с.

3. ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»