Лабораторная работа № 9  
на тему  
**«Запись одномерных массивов в редактор Microsoft Word**

**и программу Блокнот»**

# СОДЕРЖАНИЕ

СОДЕРЖАНИЕ 2

1 Задание 3

2 Основная часть 3

2.1 Блок-схемы 3

2.2 Исходный код программы 5

2.2.1 Библиотека 5

2.2.2 Основная программа 8

3 Заключение 9

# 1 Задание

Дан динамический целочисленный массив. Элементы массива могут принимать целые значения от 0 до 1000 включительно. Опишите алгоритм, который подсчитывает количество пар соседних элементов, разница между которыми не кратна 7. Гарантируется, что в массиве есть хотя бы один элемент, кратный 7.

Разработайте функцию, которая формирует новый массив из тех элементов исходного массива, значения которых делится на найденное число без остатка и выведите исходный и результирующий массивы в документ Microsoft Word и текстовый документ.

Для выполнения данного задания необходимо дополнить DLL-библиотеку из прошлой работы следующими функциями:

1. write\_to\_word(int\* array, int length), которая создаёт документ Microsoft Word и записывает массив в формате таблицы;
2. write\_to\_notepad(DataGridView^ view1, DataGridView^ view2), которая создаёт текстовый документ и в форматированном виде выводит массиы.

Также необходимо разработать событийную функцию button\_fill\_click(..), обрабатывающую нажатие кнопки «ЗАПОЛНИТЬ», которая открывает документы Microsoft Word и выводит в них полученные массивы, а так же создаёт текстовый документ и выводит в него полученные массивы в форматированном виде.

# 2 Основная часть

## 2.1 Блок-схемы

Блок-схемы, реализованные в тексте программы, указаны ниже.

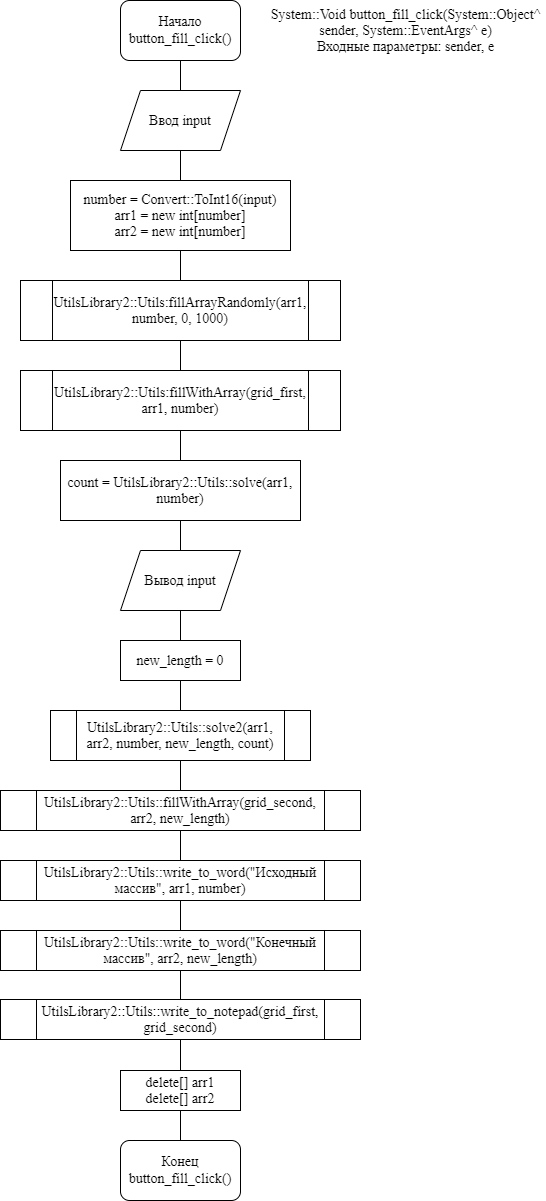


Рисунок 1 — Функция button\_ fill\_click()

## 2.2 Исходный код программы

### 2.2.1 Библиотека

#include <cstdlib>

#include "UtilsLibrary2.h"

void UtilsLibrary2::Utils::fillArrayRandomly(int\* array, size\_t length, int min, int max)

{

// Найдено ли один элемент, делящийся на 7 без остатка

bool found = false;

for (size\_t i = 0; i < length; i++)

{

// Случайное число в промежутке [min;max]

int value = rand() % (max - min + 1) + min;

array[i] = value;

if (value % 7 == 0) found = true;

}

if (!found)

{

// Принудительно устанавливаем значение

array[length / 2] = 7;

}

}

void UtilsLibrary2::Utils::fillWithArray(DataGridView^ grid, int\* array, size\_t length)

{

grid->ColumnCount = length;

grid->RowCount = 2;

for (size\_t i = 0; i < length; i++)

{

grid->Rows[0]->Cells[i]->Value = "[" + i + "]";

grid->Rows[1]->Cells[i]->Value = array[i];

}

int sum = 0;

for (int i = 0; i < grid->ColumnCount; i++)

sum += grid->Columns[i]->Width;

if (sum > 410)

grid->Width = 410;

else

grid->Width = sum;

}

size\_t UtilsLibrary2::Utils::solve(int\* array, size\_t length)

{

size\_t count = 0;

for (size\_t i = 0; i < length - 1; i++)

{

int delta = array[i] - array[i + 1];

// результат оператора % будет неверным, если delta < 0;

// остаток от деления всегда положителен с математической точки зрения

// и записывается в виде (a % b + b) % b

if ((delta % 7 + 7) % 7 != 0) count++;

}

return count;

}

void UtilsLibrary2::Utils::solve2(int\* array, int\* new\_array, int length, int& new\_length, int number)

{

int index = 0;

for (size\_t i = 0; i < length; i++)

{

if (array[i] % number == 0)

{

new\_array[index] = i;

index++;

}

}

new\_length = index;

}

void UtilsLibrary2::Utils::create\_db()

{

ADOX::Catalog^ catalog = gcnew ADOX::CatalogClass();

try

{

catalog->Create("Provider = Microsoft.Jet.OLEDB.4.0; Data Source = solution.mdb");

MessageBox::Show("Файл базы данных успешно создан", "Инфо", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

catch (System::Runtime::InteropServices::COMException^ exception)

{

MessageBox::Show(exception->Message, "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

finally

{

catalog = nullptr;

}

}

void UtilsLibrary2::Utils::configure\_db()

{

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source = solution.mdb");

p->Open();

auto c = gcnew OleDbCommand("CREATE TABLE [Результат]([Индекс] counter, [Исходный массив] char(200)," +

"[Результирующий массив] char(200))", p);

try

{

c->ExecuteNonQuery();

MessageBox::Show("Структура базы данных успешно сформирована", "Инфо", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

catch (OleDbException^ exception)

{

MessageBox::Show(exception->Message, "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

p->Close();

}

void UtilsLibrary2::Utils::write\_to\_db(int\* array, int\* new\_array, int length, int new\_length)

{

for (size\_t i = 0; i < length; i++)

{

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source = solution.mdb");

p->Open();

if (i < new\_length)

{

auto c = gcnew OleDbCommand("INSERT INTO[Результат](" +

"[Исходный массив], [Результирующий массив]) VALUES('" + array[i] + "','" + new\_array[i] + " ')");

c->Connection = p;

c->ExecuteNonQuery();

}

else

{

auto c = gcnew OleDbCommand("INSERT INTO [Результат](" +

"[Исходный массив], [Результирующий массив]) VALUES('" + array[i] + "','')");

c->Connection = p;

c->ExecuteNonQuery();

}

}

}

void UtilsLibrary2::Utils::write\_to\_word(String^ title, int\* array, int length)

{

auto word = gcnew Microsoft::Office::Interop::Word::ApplicationClass();

word->Visible = true;

auto missing = Type::Missing;

auto document = word->Documents->Add(missing, missing, missing, missing);

Object^ behavior1 = Microsoft::Office::Interop::Word::WdDefaultTableBehavior::wdWord9TableBehavior;

Object^ behavior2 = Microsoft::Office::Interop::Word::WdAutoFitBehavior::wdAutoFitContent;

word->Selection->TypeText(title);

Microsoft::Office::Interop::Word::Table^ table = word->ActiveDocument->Tables->Add(word->Selection->Range,

2, length, behavior1, behavior2);

for (size\_t i = 0; i < length; i++)

{

table->Cell(1, i + 1)->Range->InsertAfter("[" + Convert::ToString(i) + "]");

String^ str = String::Format("{0:f0}", array[i]);

table->Cell(2, i + 1)->Range->InsertAfter(str);

}

}

void UtilsLibrary2::Utils::write\_to\_notepad(DataGridView^ view1, DataGridView^ view2)

{

String^ f = Microsoft::VisualBasic::Interaction::InputBox("Введите путь по которому будет сохранен файл и его имя. " +

"Расширение добавляется автоматически\nНапример:C:\\Users\\name\\repos\'имя файла'", "Ввод", "Ввод", -1, -1);

String^ filename = f + ".txt";

try {

auto encoding = System::Text::Encoding::GetEncoding(1251);

auto stream = gcnew IO::StreamWriter(filename, false, encoding);

stream->WriteLine("Исходный массив | Результирующий массив");

const char\* tst = "Исходный массив |";

for (int i = 0; i < view1->ColumnCount; i++)

{

if (i < view2->ColumnCount)

stream->WriteLine(String::Format("{0, 18}",

Convert::ToString(view1->Rows[1]->Cells[i]->Value) + " | ") +

Convert::ToString(view2->Rows[1]->Cells[i]->Value));

else

stream->WriteLine(String::Format("{0, 17}",

Convert::ToString(view1->Rows[1]->Cells[i]->Value) + " |"));

}

stream->Close();

MessageBox::Show("Запись произведена успешна");

}

catch (System::Exception^ exception) {

MessageBox::Show(exception->Message, "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

}

### 2.2.2 Основная программа

System::Void button\_fill\_click(System::Object^ sender, System::EventArgs^ e)

{

button\_fill->Enabled = false;

String^ input = Interaction::InputBox("Введите количество элементов массива", "Ввод", "10", -1, -1);

int number = Convert::ToInt16(input);

int\* arr1 = new int[number];

int\* arr2 = new int[number];

UtilsLibrary2::Utils::fillArrayRandomly(arr1, number, 0, 1000);

UtilsLibrary2::Utils::fillWithArray(grid\_first, arr1, number);

size\_t count = UtilsLibrary2::Utils::solve(arr1, number);

MessageBox::Show("Количество пар соседних элементов, разница\nмежду которыми НЕ кратна 7: " +

Convert::ToString(count), "Вывод", MessageBoxButtons::OK, MessageBoxIcon::Information);

int new\_length = 0;

UtilsLibrary2::Utils::solve2(arr1, arr2, number, new\_length, count);

UtilsLibrary2::Utils::fillWithArray(grid\_second, arr2, new\_length);

UtilsLibrary2::Utils::write\_to\_word("Исходный массив", arr1, number);

UtilsLibrary2::Utils::write\_to\_word("Конечный массив", arr2, new\_length);

UtilsLibrary2::Utils::write\_to\_notepad(grid\_first, grid\_second);

delete[] arr1;

delete[] arr2;

button\_fill->Enabled = true;

}

# 3 Заключение

Техническое задание были успешно реализовано на языке программирования Visual C++. Программа работает корректно, полученные результаты, представленные на рисунках ниже, верны.

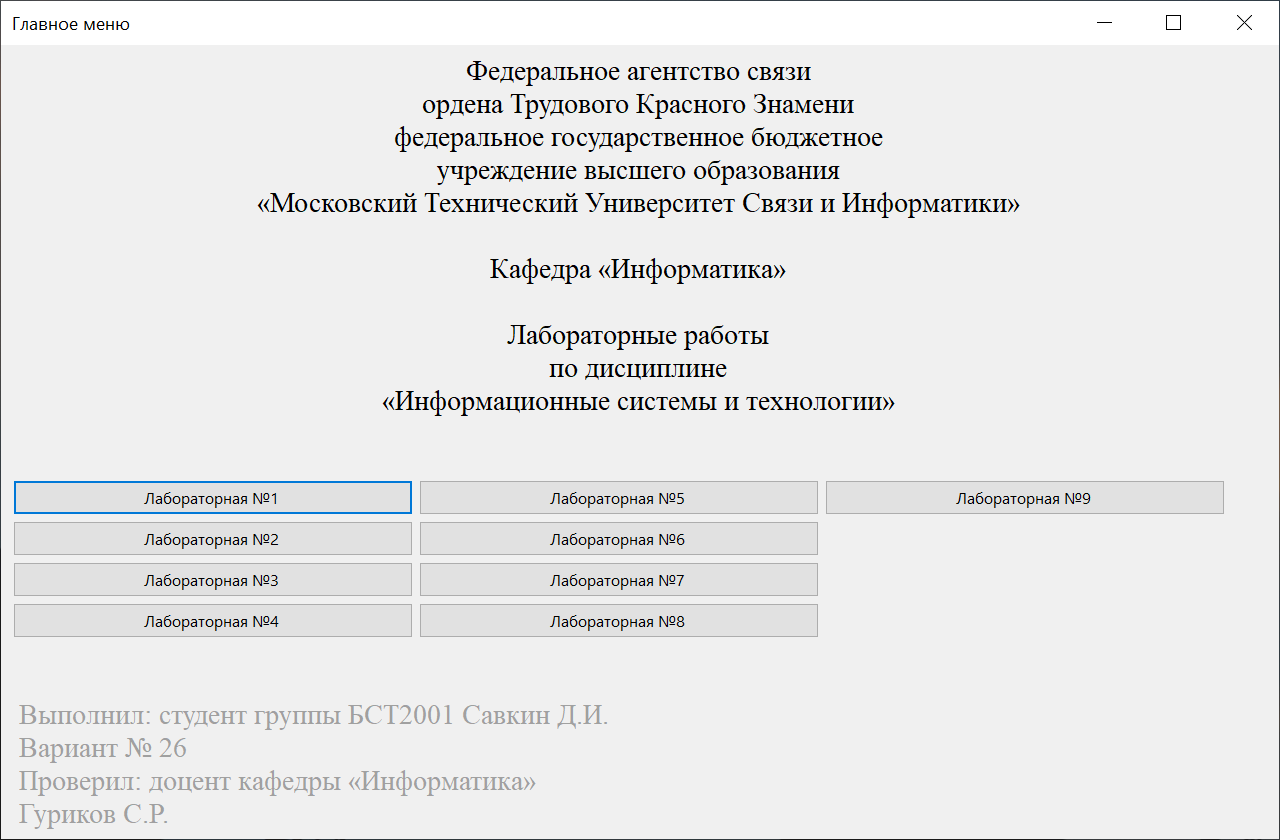


Рисунок 2 — Главное меню программы

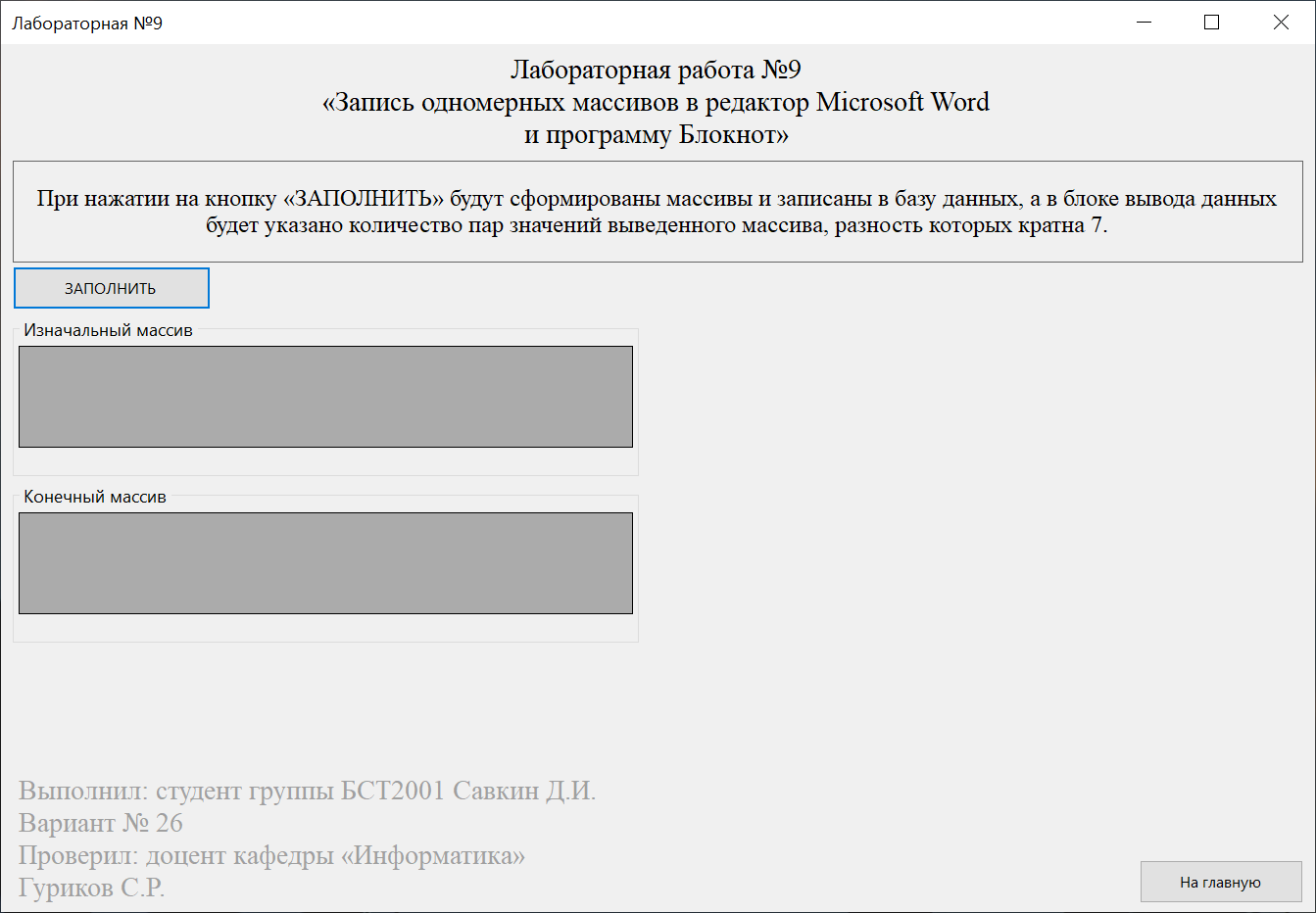


Рисунок 3 — Форма лабораторной работы

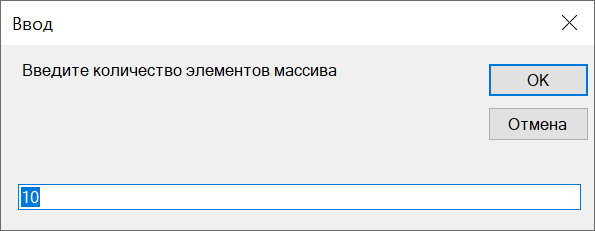


Рисунок 4 — Организация ввода размера массива

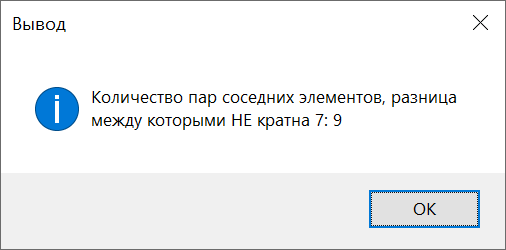


Рисунок 5 — Организация вывода результатов

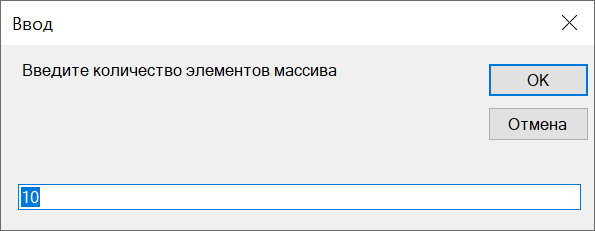


Рисунок 6 — Организация ввода количества элементов

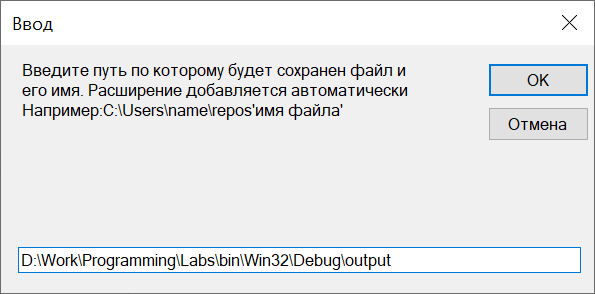


Рисунок 7 — Ввод имени текстового файла

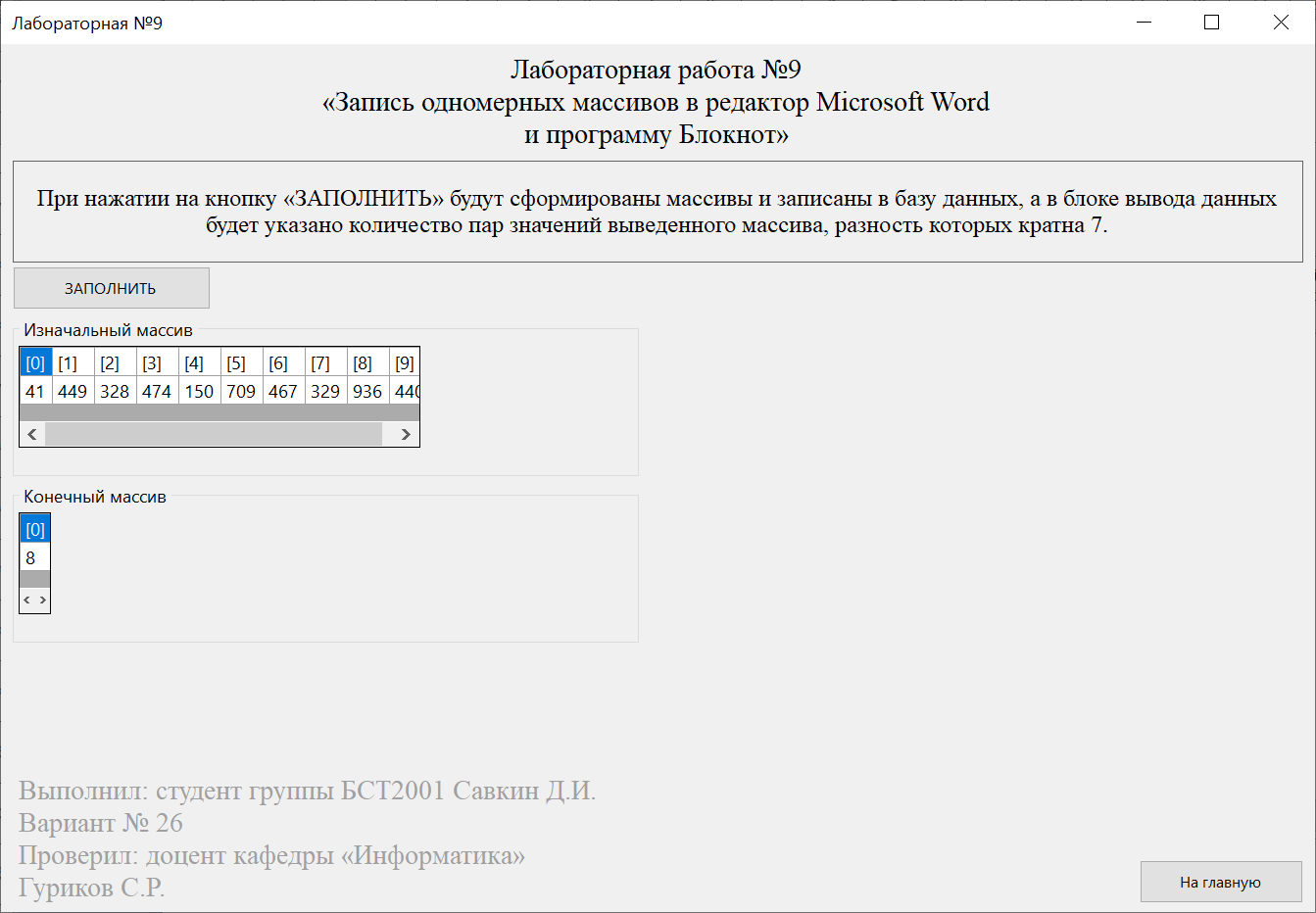


Рисунок 8 — Вывод результатов

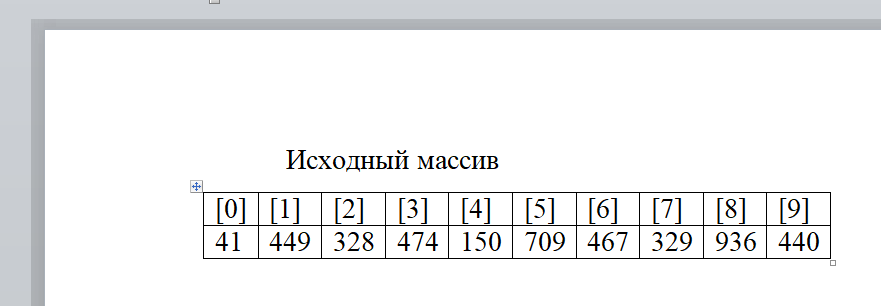


Рисунок 9 — Вывод исходного массива в документ

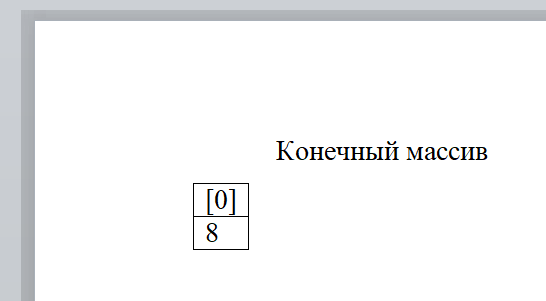


Рисунок 10 — Вывод конечного массива в документ

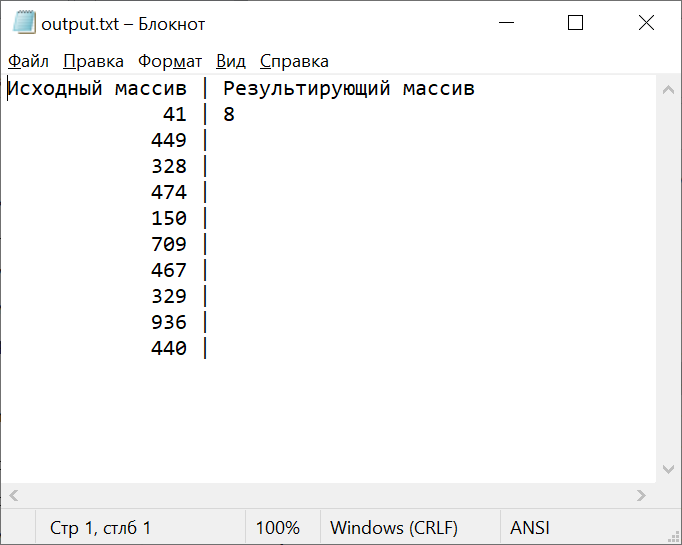


Рисунок 11 — Вывод массивов в текстовый файл