Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рязанский государственный радиотехнический университет

имени В. Ф. Уткина»

Кафедра «Вычислительная и прикладная математика»

**Отчет**

**по лабораторной работе № 1**

**по дисциплине**

**«Объектно-ориентированное программирование»**

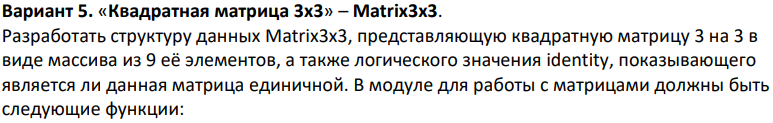
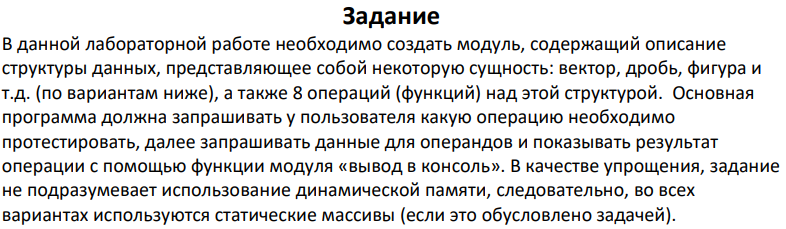
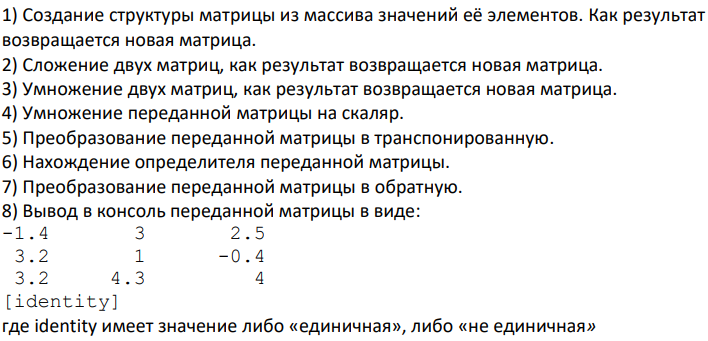
**на тему**

**«Структуры и модули»**

Выполнил:   
студент гр. 143  
Вербицкая И. С.

Проверил:  
Антипов О.В.

Рязань 2022

**Анализ задания:**

*Входные данные:*

* квадратные матрицы A, B, C с действительными элементами размером 3x3;
* целочисленная переменная Sclr ­- скаляр;
* целочисленная переменная Key – номер команды;
* целочисленные переменные Name1 и Name2 – номера матриц, с которыми производятся преобразования.

*ОДЗ:*

* 1≤Key≤9;
* 1≤Name1≤3;
* 1≤Name2≤3.

*Выходные данные*:

* квадратные матрицы A, B, C с действительными элементами размером 3x3 и идентификатором единичности id;
* определитель выбранной матрицы;
* сообщения об ошибках (при попытке ввести некорректный номер команды/матрицы или при попытке преобразовать матрицу с нулевым определителем в обратную).

*Этапы решения задачи*:

* составить алгоритм работы программы;
* составить программу;
* провести проверку работы программы.

**Описание структуры данных:**

В подпрограмме используется структура Matrix3x3, представляющая собой квадратную матрицу размером 3 на 3, состоящая из следующих полей:

* двумерный массив действительных чисел размерностью 3 на 3;
* логическое значение id, показывающее, является матрица единичной (1) или нет (0).

**Подпрограммы для работы со структурой данных:**

**bool Ident (Matrix3x3 Matr):**

* функция по определению поля id в структуре матрицы;
* входной параметр: Matrix3x3 Matr – структура данных (*здесь и далее*);
* функция возвращает 0 если матрица не единичная и 1 – если единичная;

**Matrix3x3 Nulls (Matrix3x3 Matr):**

* заполнение матрицы нулями;
* функция возвращает структуру матрицы.

**Matrix3x3 Create (Matrix3x3 New):**

* создание структуры матрицы;
* входной параметр: Matrix3x3 New – структура данных;
* функция возвращает структуру матрицы.

**Matrix3x3 Summ (Matrix3x3 A, Matrix3x3 B):**

* сложение двух матриц;
* входные параметры: Matrix3x3 A и Matrix3x3 B – структуры данных;
* функция возвращает структуру матрицы.

**Matrix3x3 Mult (Matrix3x3 A, Matrix3x3 B):**

* умножение двух матриц;
* входные параметры: Matrix3x3 A и Matrix3x3 B – структуры данных;
* функция возвращает структуру матрицы.

**void Scalar (Matrix3x3\* Matr, int n):**

* умножение матрицы на скаляр;
* входные параметры: Matrix3x3\* Matr – указатель на структуру данных (*здесь и далее*); int n – целочисленное значение скаляра;
* подпрограмма модифицирует структуру данных по входному указателю.

**void Transpos (Matrix3x3\* Matr):**

* транспонирование матрицы;
* подпрограмма модифицирует структуру данных по входному указателю.

**float Det (Matrix3x3 Matr):**

* нахождение определителя матрицы;
* функция возвращает действительное значение определителя переданной матрицы.

**bool Reverse (Matrix3x3\* Matr):**

* преобразование матрицы в обратную;
* подпрограмма модифицирует структуру данных по входному указателю.

**void Out (Matrix3x3 Matr, int n):**

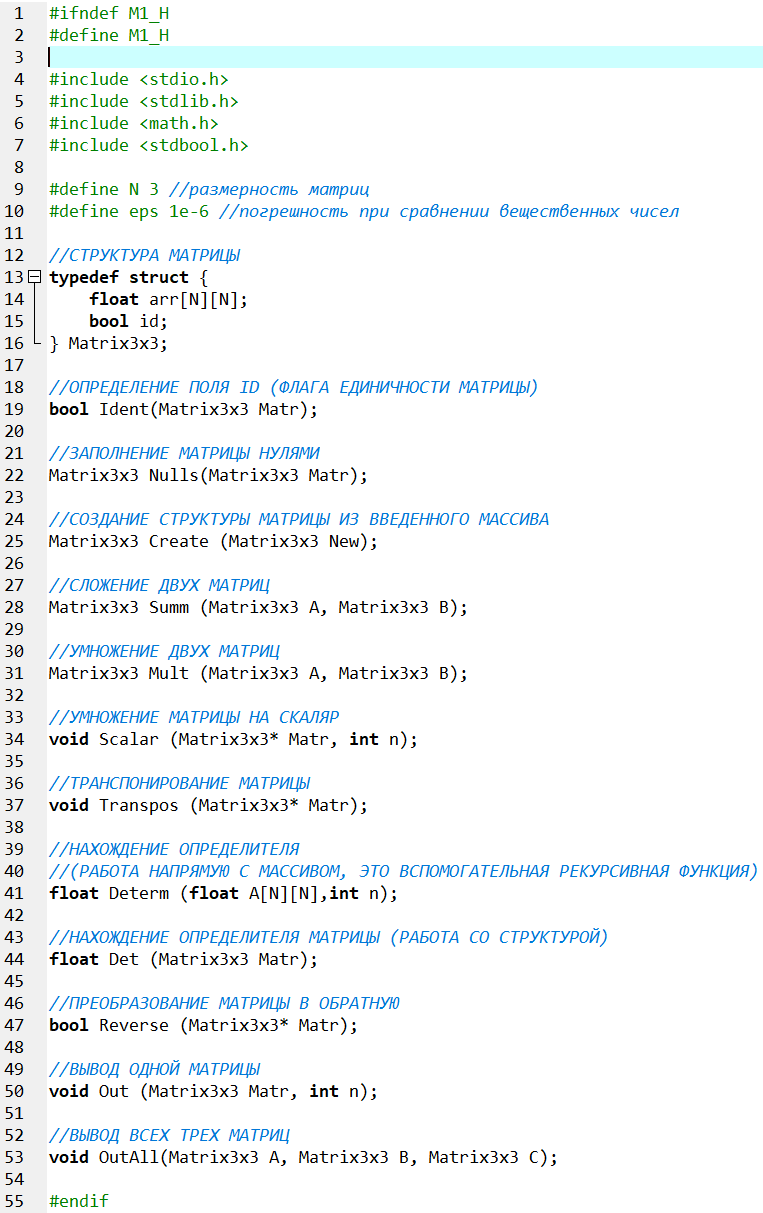
* вывод матрицы в консоль;
* входные параметры: int n – номер матрицы (если 1, матрица А; если 2 – В; 3 – С);

**void OutAll (Matrix3x3 A, Matrix3x3 B, Matrix3x3 C):**

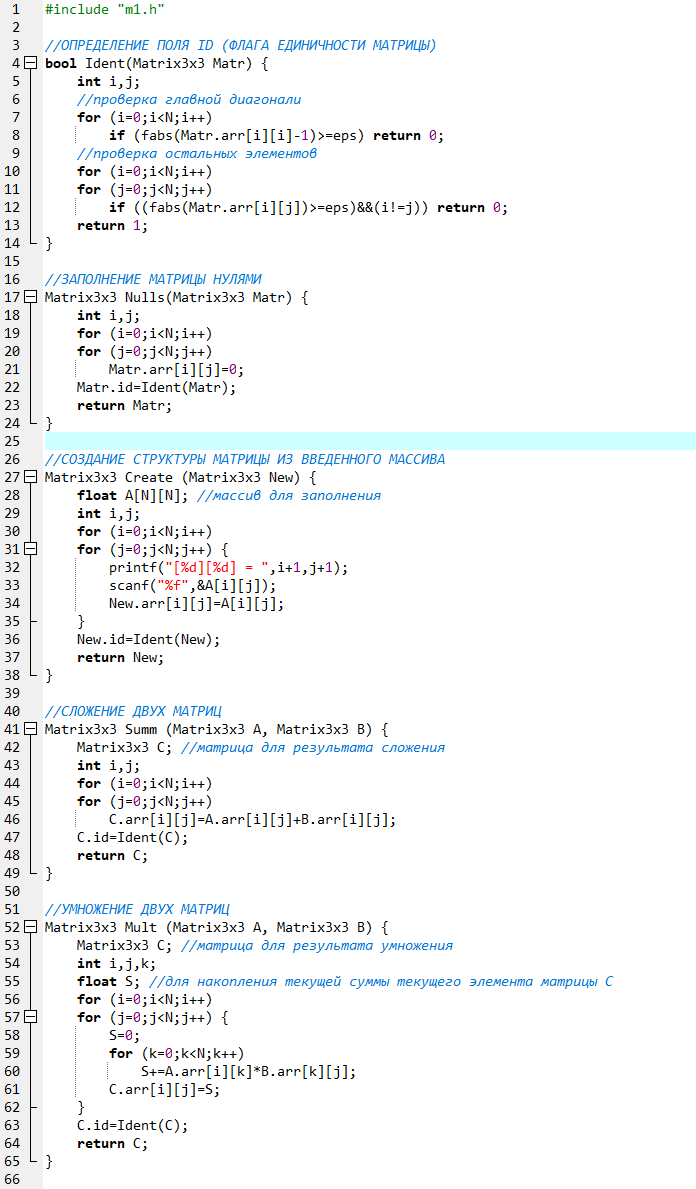
* вывод трёх матриц в консоль;
* входные параметры: Matrix3x3 A, Matrix3x3 B, Matrix3x3 C – структуры данных.

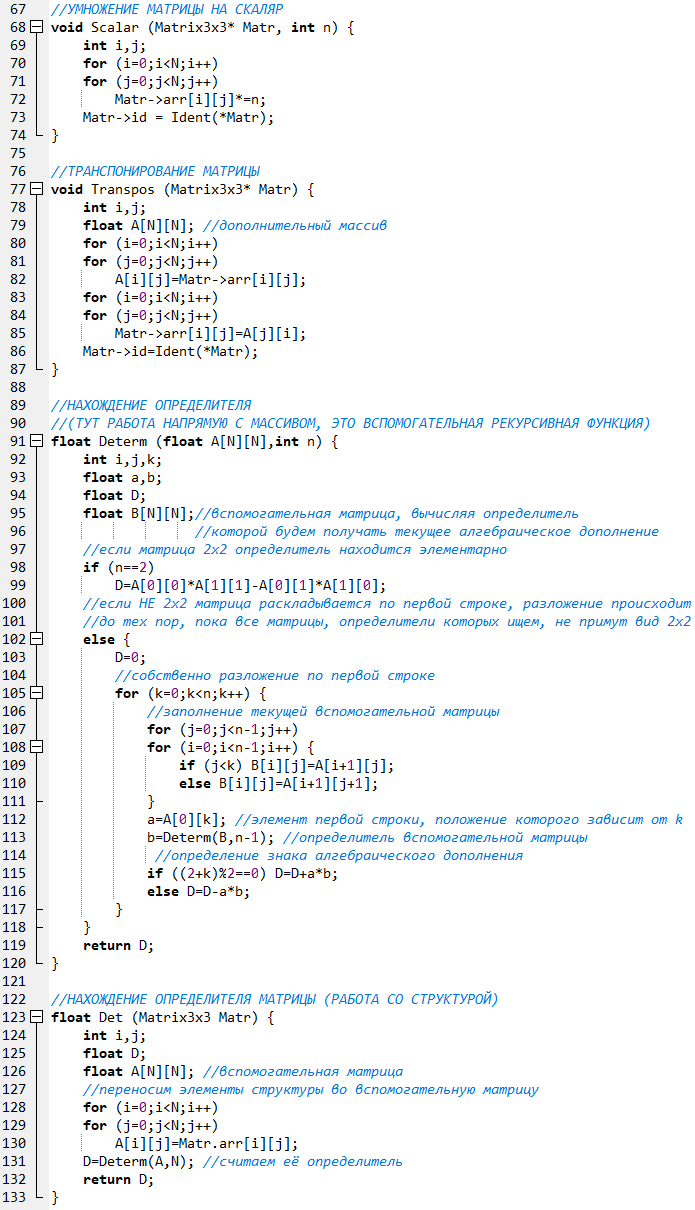
**Листинг программы:**

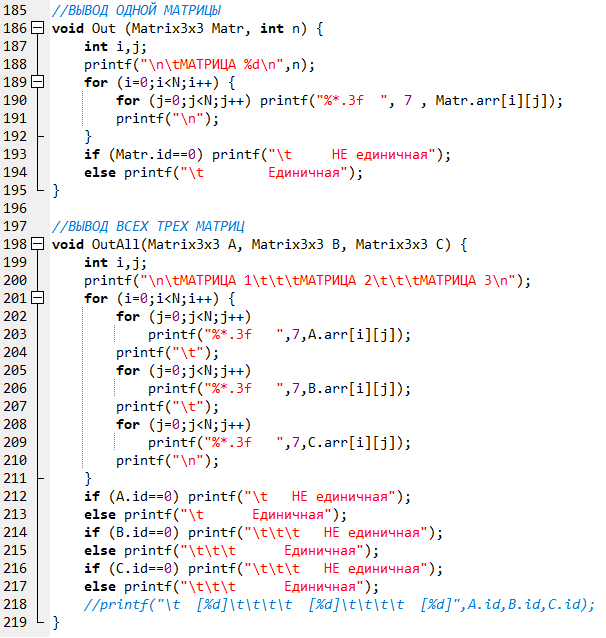
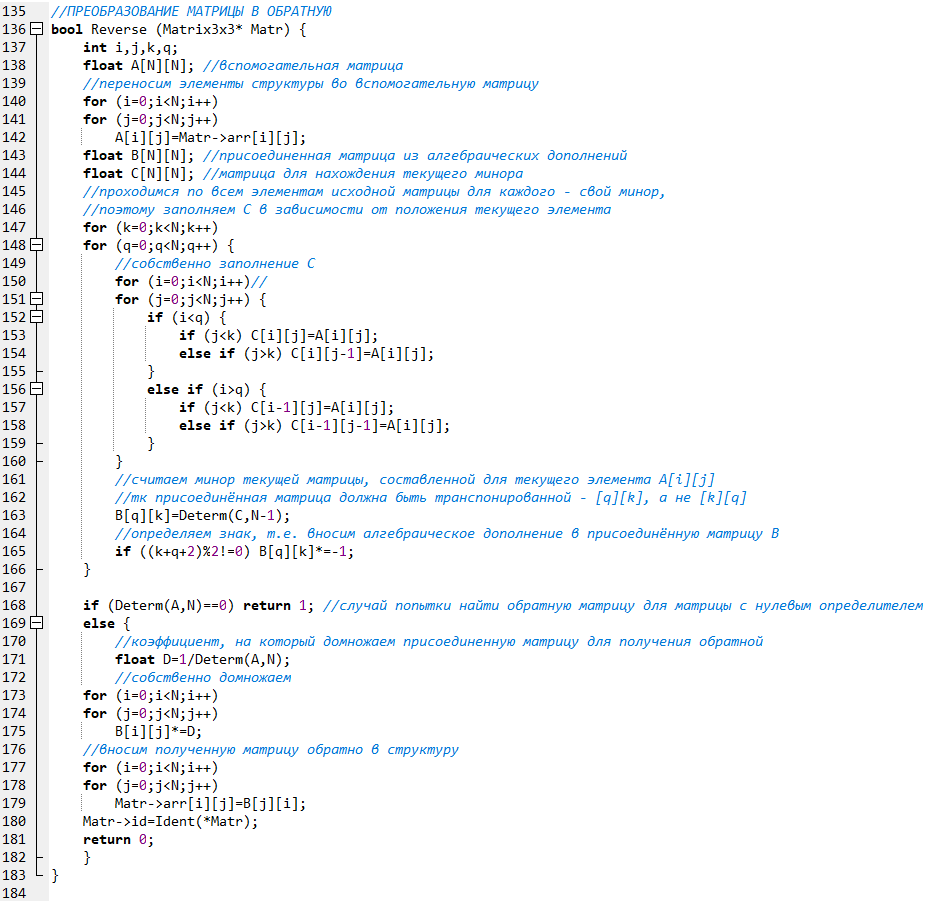
*Файл m1.h*



*Файл m1.c*

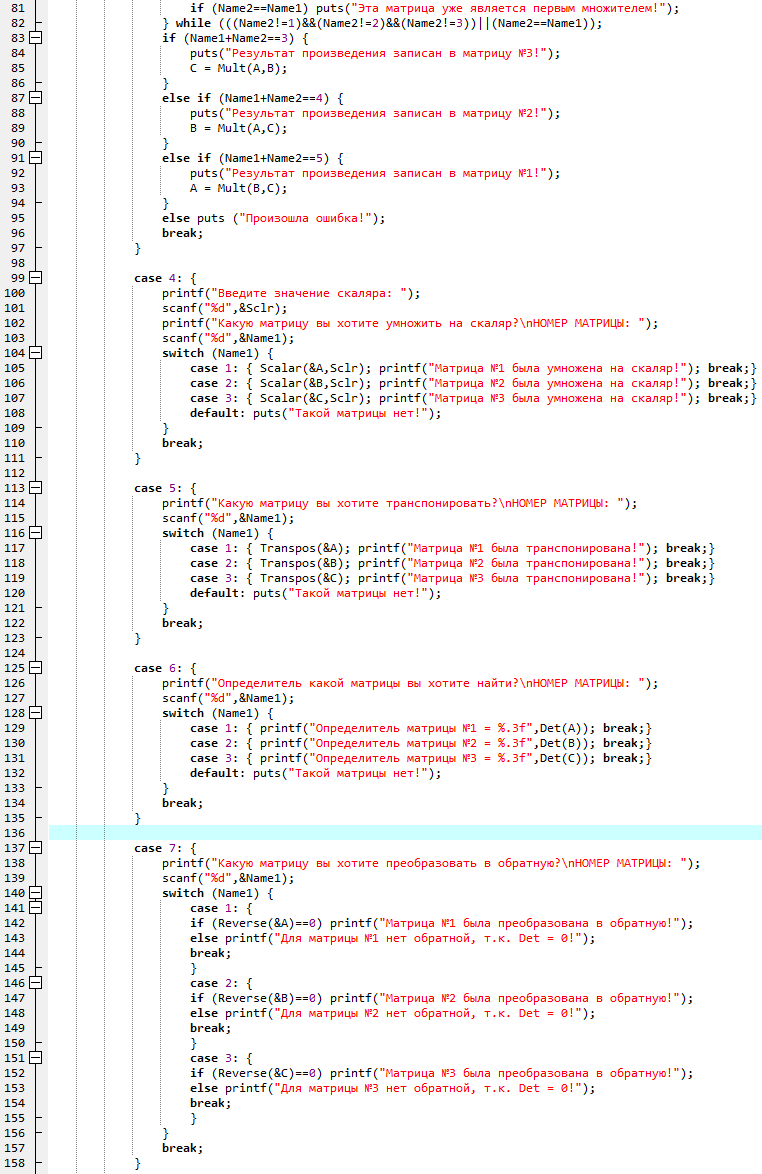


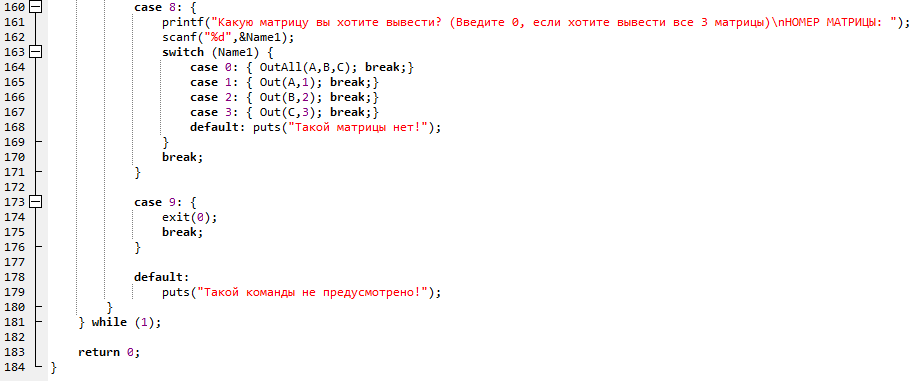




*Файл main.c*







**Результаты работы программы и проверка:**

*Тест №1*

Для демонстрации работоспособности программы зададим значения ячеек матрицы №1 и выведем её (рисунок 1). Заданные значения были записаны в матрицу без искажений.

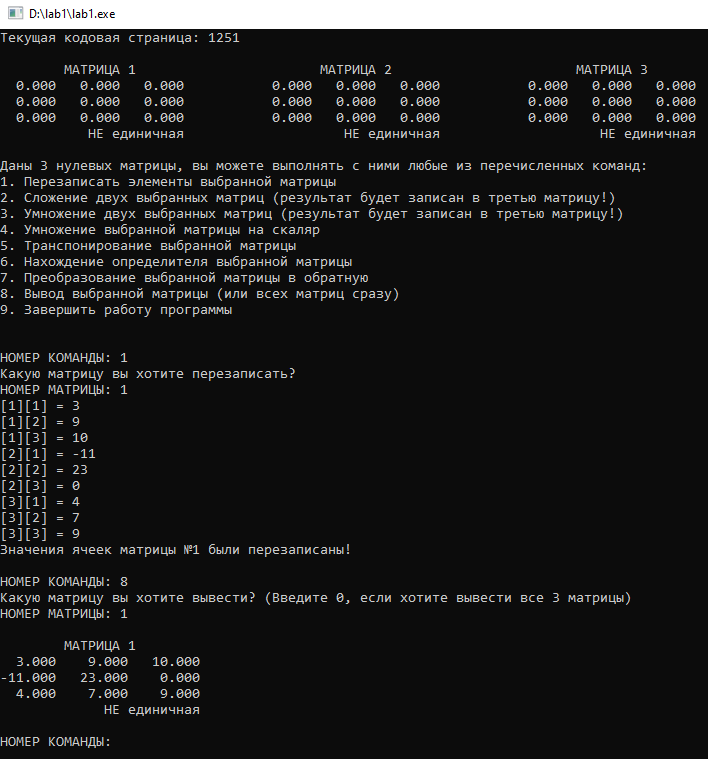


Рисунок 1 – создание структуры матрицы из массива значений её элементов

Зададим так же значения ячеек матрицы №2. Затем сложим матрицу №1 и матрицу №2. В результате будет получена новая матрица, записанная в матрицу №3 (рисунок 2). Правильность сложения проконтролируем с помощью программы Excel (рисунок 3).

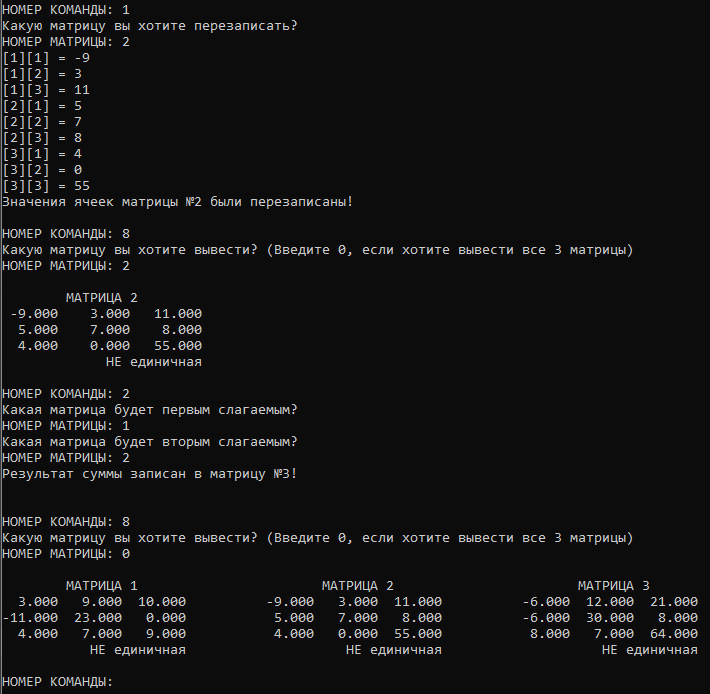


Рисунок 2 – сложение двух матриц

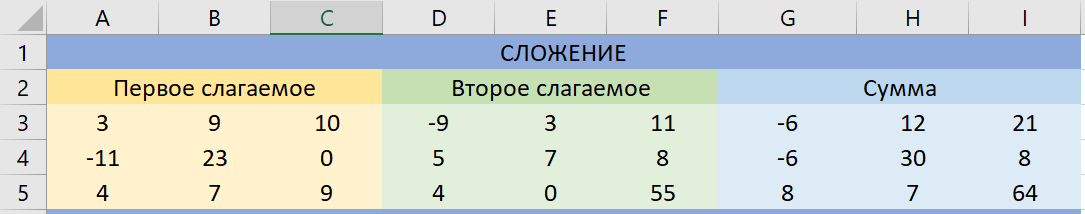


Рисунок 3 – проверка функции сложения матриц

Затем умножим матрицу №1 на матрицу №2. В результате умножения в матрицу №3 будет записан результат операции (рисунок 4). Результат произведения матриц совпал с результатом, полученным с помощью Excel (рисунок 5).

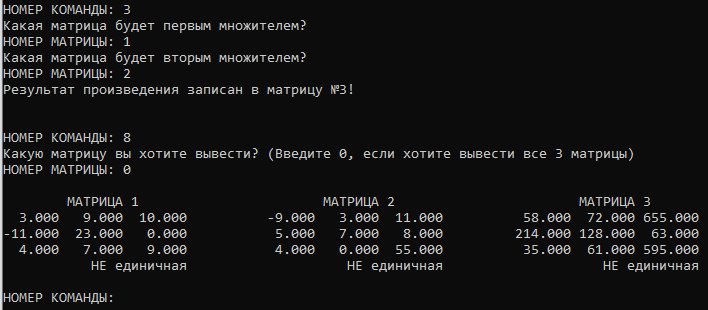


Рисунок 4 – умножение двух матриц

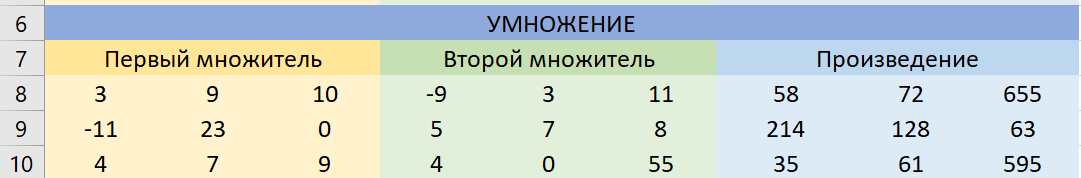


Рисунок 5 – проверка функции умножения матриц

Затем умножим матрицу №2 на скаляр. В результате значения ячеек матрицы будут изменены (рисунок 6). Проверка в Excel подтвердила правильность работы функции (рисунок 7).

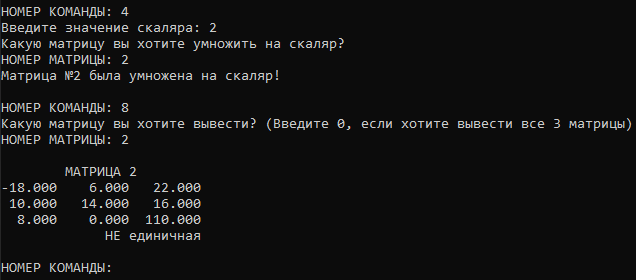


Рисунок 6 – умножение матрицы на скаляр



Рисунок 7 – проверка функции умножения матрицы на скаляр

Протестируем также процедуру транспонирования матрицы, выбрав команду, транспонирующую матрицу №1 (рисунок 8) и осуществив проверку правильности с помощью Excel (рисунок 9).

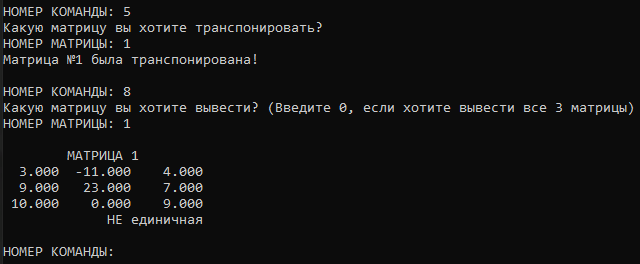


Рисунок 8 – транспонирование матрицы



Рисунок 9 – проверка процедуры транспонирования матрицы

Найдём определитель матрицы №1 (рисунок 10) и так же проверим правильность с помощью Excel (рисунок 11).

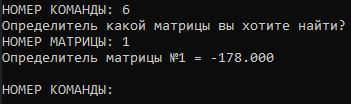


Рисунок 10 – нахождение определителя

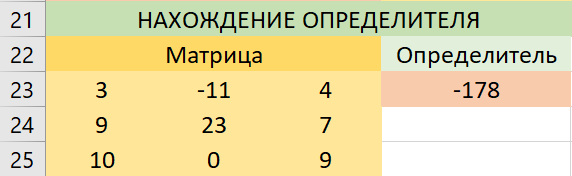


Рисунок 11 – проверка функции нахождения определителя

Преобразуем матрицу №3 в обратную. Полученная в результате изменённая матрица №3 (рисунок 12) совпадает с матрицей, полученной с помощью программы Excel (рисунок 13).

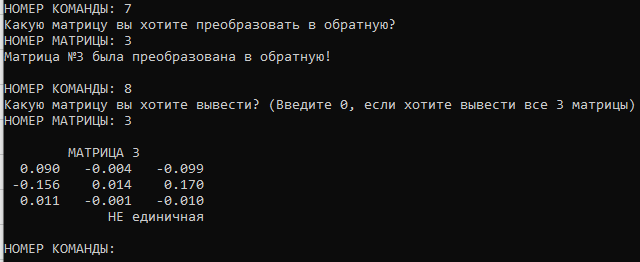


Рисунок 12 – преобразование матрицы в обратную



Рисунок 13 – проверка процедуры преобразования матрицы в обратную

Наконец, убедимся в корректности работы процедуры вывода матриц, выведя сначала каждую из трех матриц по отдельности, а затем – все три вместе; после чего завершим работу программы соответствующим номером команды (рисунок 14). Заметим, что матрицы сохранили свои значения, полученные в результате применения всех вышеописанных процедур.

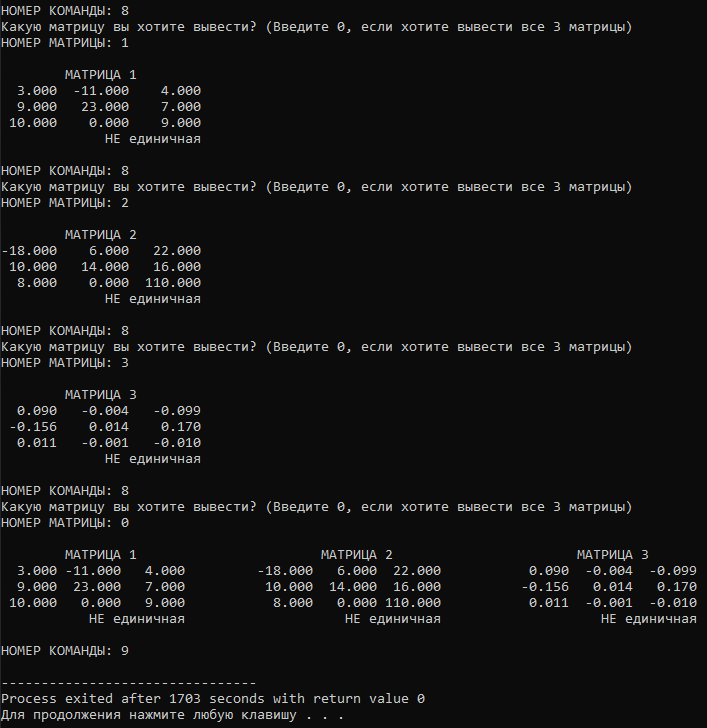


Рисунок 14 – вывод матриц

*Тест №2*

Альтернативный способ проверки правильности работы двух функций программы (умножения матриц и нахождения обратной матрицы) – соблюдение равенства из линейной алгебры, где – исходная матрица, - обратная ей, – единичная матрица. С помощью программы зададим одинаковые матрицы №1 и №2, затем преобразуем матрицу №2 в обратную и умножим 1-ую матрицу на 2-ую. Как и ожидалось, матрица №3 – единичная (рисунок 15) – как результат произведения матрицы №1 на матрицу №2, что подтверждает корректность работы программы, ведь вышеуказанное тождество было соблюдено.

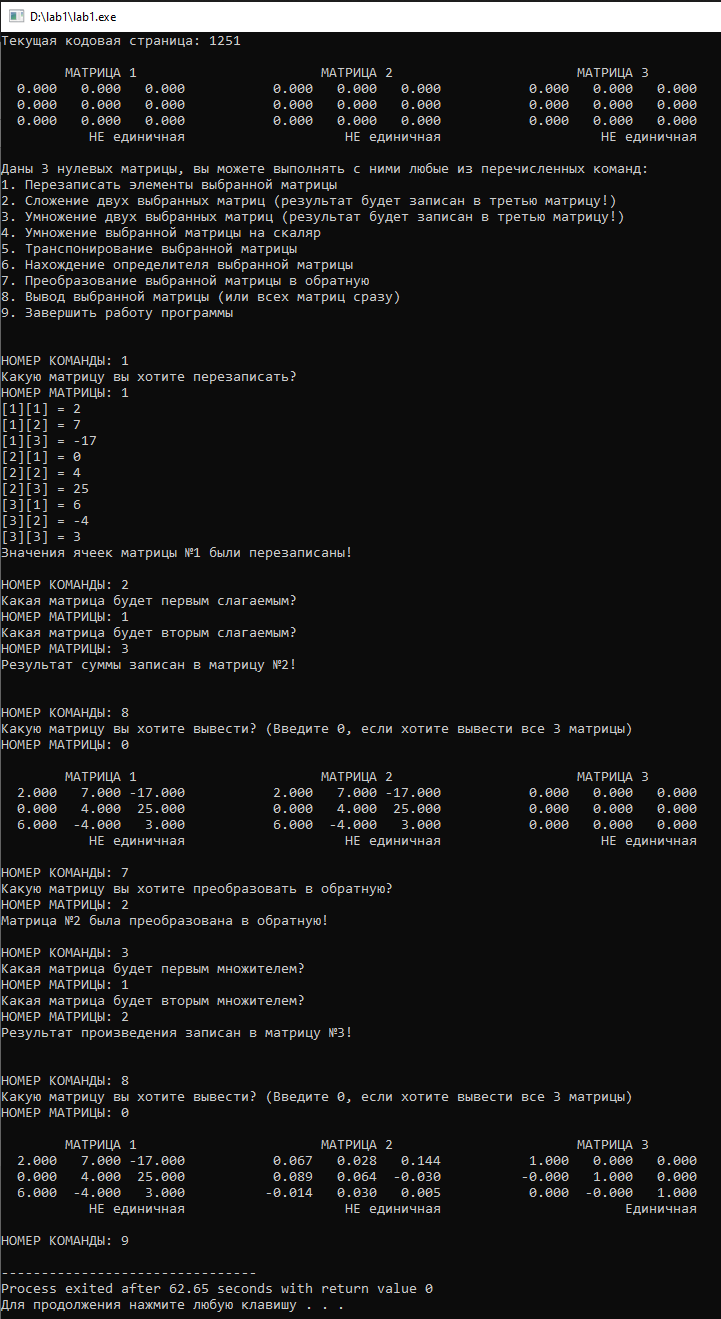


Рисунок 15