МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Университет ИТМО

Факультет систем управления и робототехники

ЛАБОТАТОРНАЯ РАБОТА № 1

по дисциплине «Практическая линейная алгебра»

Выполнила:

Студентка группы R3281

Троицкая Тамара Андреевна

Преподаватель: Перегудин Алексей Алексеевич

Санкт-Петербург, 2023

Приветствую тебя, уважаемый читатель моего отчёта. Он обещает быть обширным, так что перед началом я посоветовала бы тебе заварить кофе или чего покрепче.

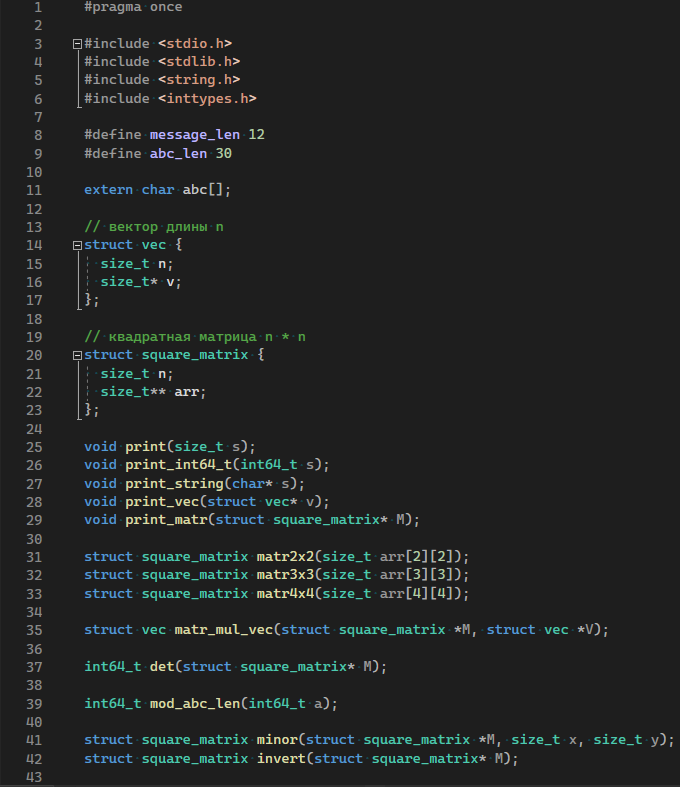
Код для лабы я писала на Си, поэтому написала вручную все структуры и функции, потребовавшиеся для работы с матрицами и векторами. Я постаралась написать комментарии к коду максимально подробно, чтобы всё было понятно даже человеку, никогда не писавшему на Си. Можете просто читать комментарии перед функциями, чтобы понять, что они делают, но реализацию я тоже приложу.

Начнём стандартно с задачи №1, в которой нужно было реализовать шифр Хилла. Говоря коротко, нужно было:

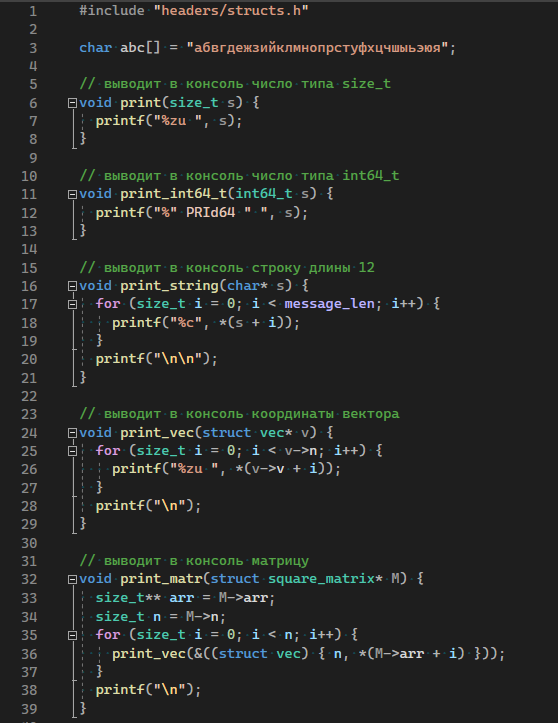
1. Придумать квадратную матрицу-ключ
2. Перевести сообщение из 12 букв в массив из 12 чисел -- номеров этих букв в заданном “алфавите”
3. Разбить массив этих чисел в массив векторов такого же размера, как матрица-ключ
4. Поочерёдно умножать матрицу-ключ на эти векторы, формируя массив векторов результата
5. Сконкатенировать, то есть склеить все векторы в один массив чисел
6. Перевести этот массив чисел обратно в буквы

Теперь, когда идея понятна, перейдём к коду. Файл structs.h. В нём объявлены функции, реализованные в файле structs.c. Эти функции можно будет использовать в любом файле, в начале которого написано #include “structs.h”. Это именно вспомогательные функции для работы с матрицами и векторами, о которых я говорила в прошлом абзаце. Скоро рассмотрим действие каждой функции более подробно.

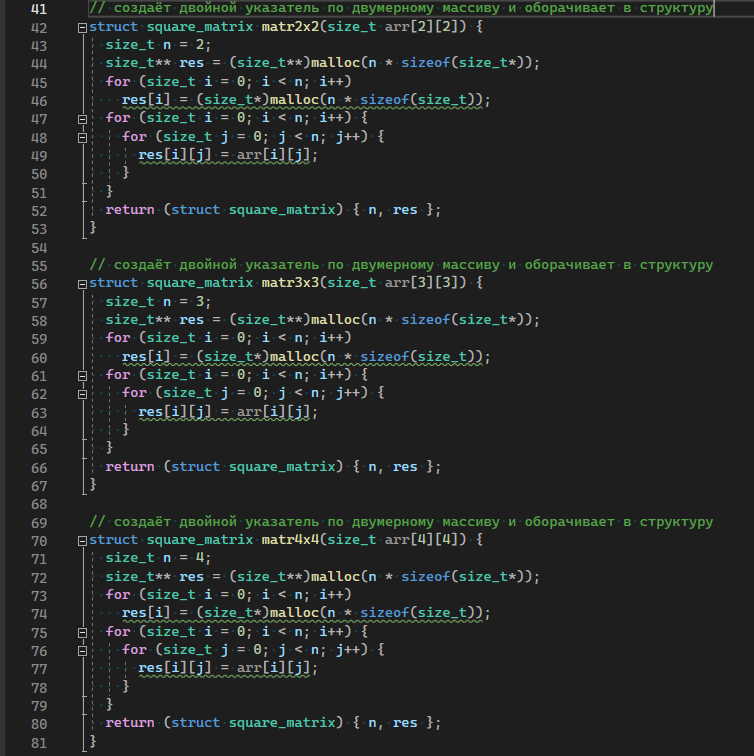
Я написала структуру для матрицы и для вектора, чтобы можно было в явном виде задавать и использовать размеры матрицы и вектора. Это было бы труднее и не так безопасно, если бы мы использовали просто указатели. Также при моём подходе можно нормально создавать указатель на матрицу и массив векторов, не путаясь в трёх звёздочках и прочем.



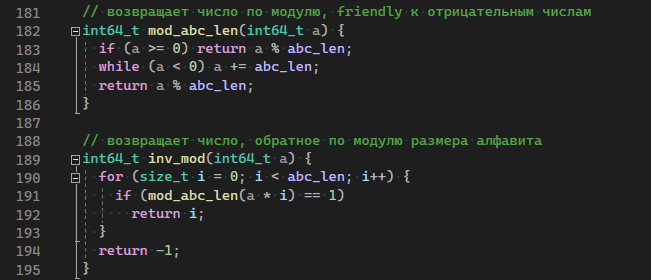
Перейдём к рассмотрению реализации функций. Файл structs.c. В начале файла -- массив используемых символов алфавита. Первые 5 функций выводят в консоль разные типы данных, это нужно для отладки и просмотра результата.

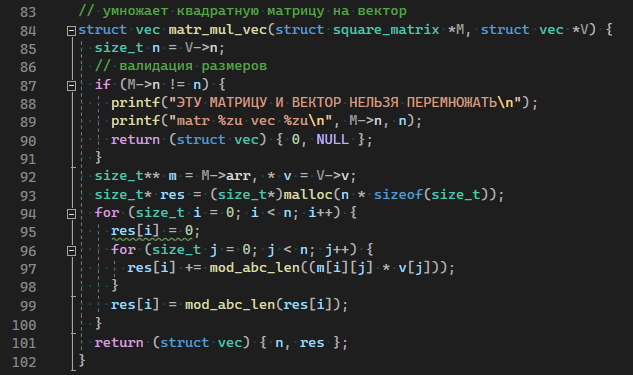


Функции для упрощённой инициализации матриц с помощью двумерного массива:

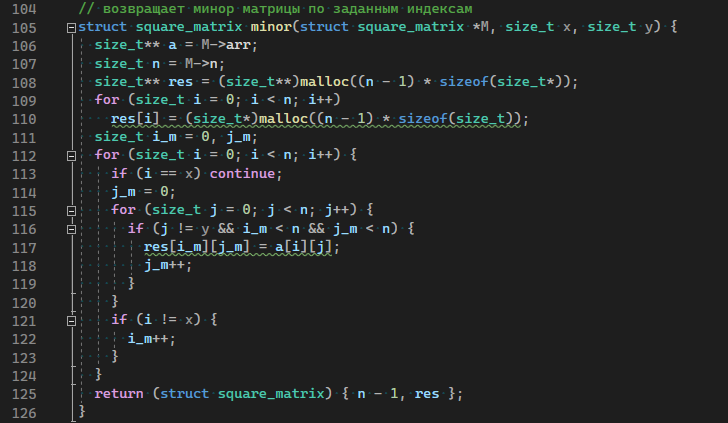


Две функции для работы в кольце (в общем, чтоб искать что-то по модулю)

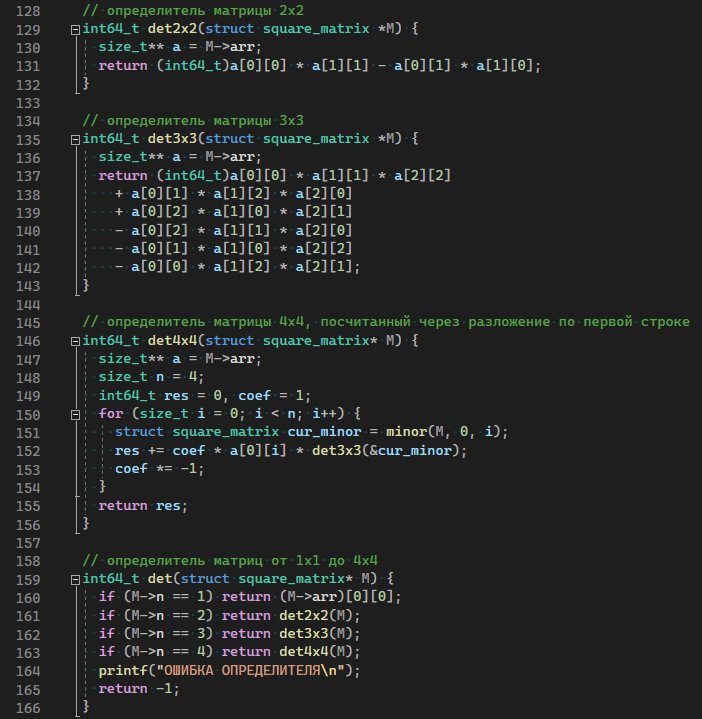




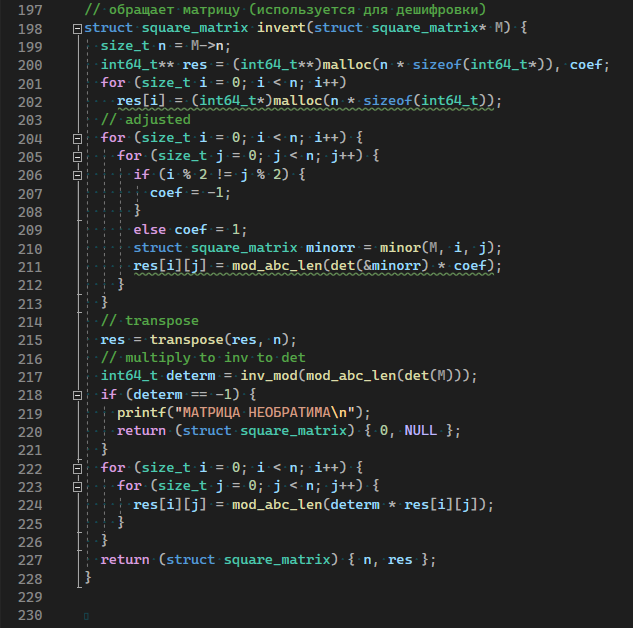
Возвращает минор



Функции для подсчёта определителя (последняя нужна для укороченного вызова первых трёх)



Напомню факт из матричной алгебры. Чтобы обратить матрицу, надо найти матрицу, состоящую из определителей её миноров, сделать матрицу алгебраических дополнений, домножив нужные элементы на -1, транспонировать её и полученную матрицу домножить на число, обратное дискриминанту, не забывая, что мы в кольце вычетов по модулю:



Итак, мы рассмотрели все вспомогательные функции работы с матрицами и векторами.

Интересно, дочитал ли кто-нибудь до этой строчки...