Лабораторная работа №3

Управляющие структуры

Дурдалыев Максат

Содержание

# 1. Введение

## 1.1 Цели и задачи

**Цель работы**

Основная цель работы — освоить применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами[1].

**Задание**

1. Используя Jupyter Lab, повторите примеры.
2. Выполните задания для самостоятельной работы[2].

# 2. Выполнение лабораторной работы

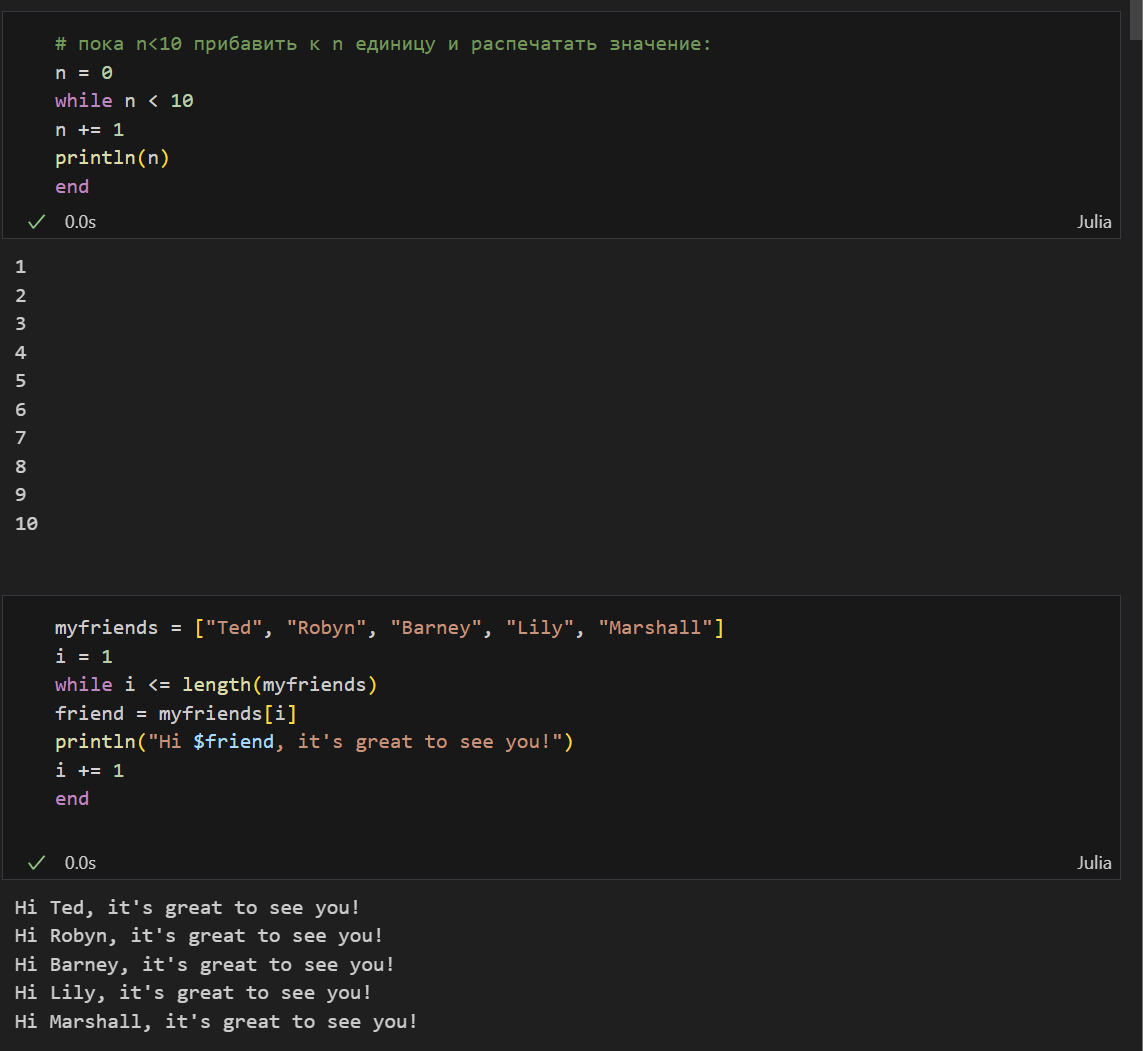
## 2.1 Циклы while и for

Для различных операций, связанных с перебором индексируемых элементов структур данных, традиционно используются циклы while и for.

Синтаксис while

while <condition>  
 <loop body>  
end

Примеры использования цикла while (**?@fig-001**):



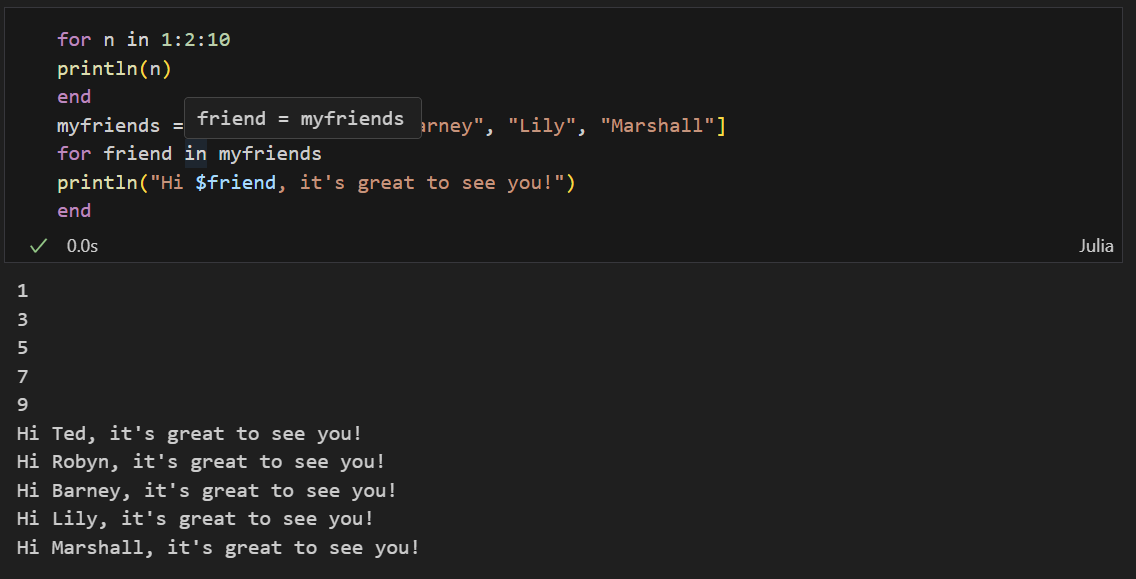
Примеры использования цикла while

Такие же результаты можно получить при использовании цикла for.

Синтаксис for

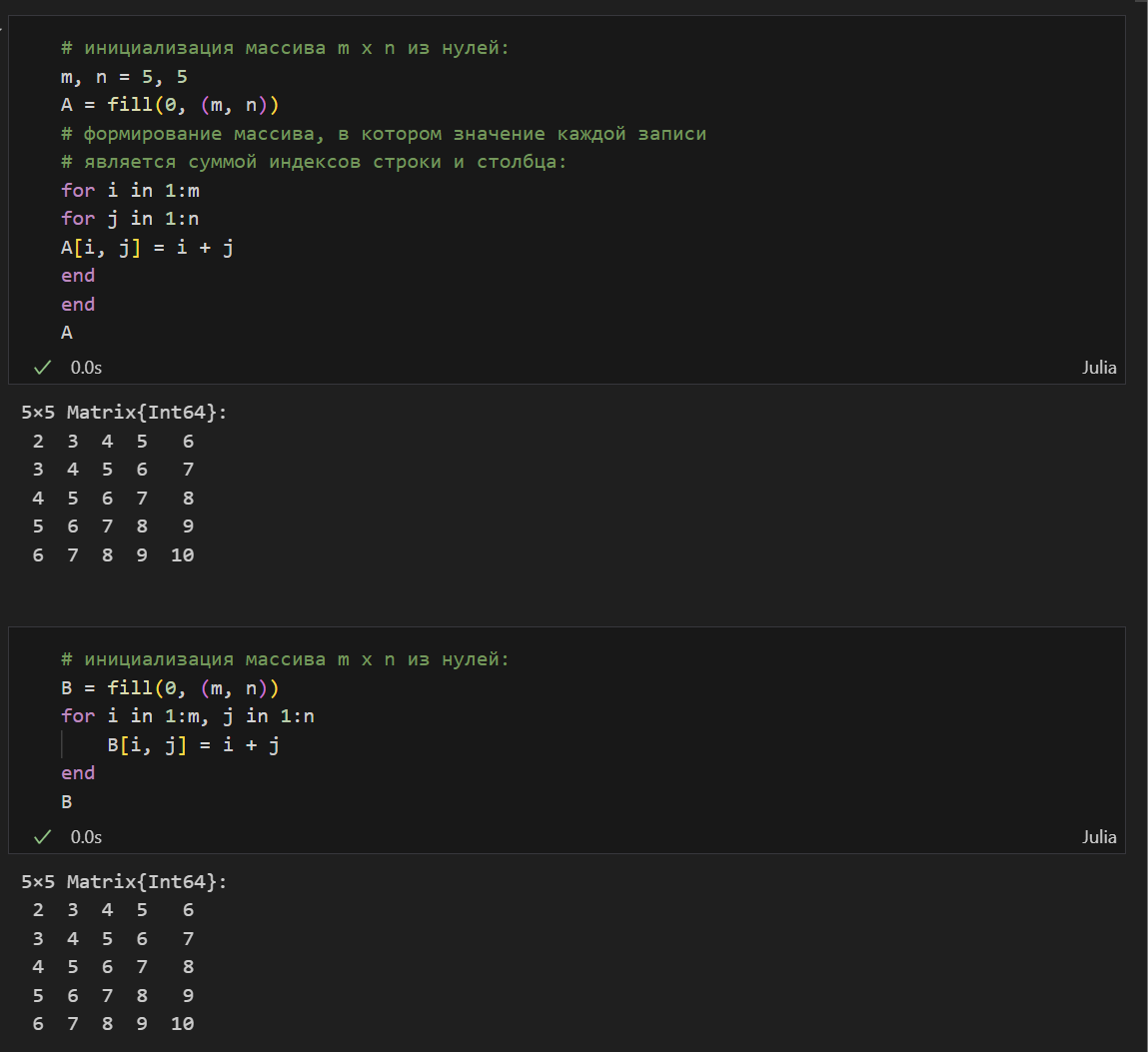
for <variable> in <range>  
 <loop body>  
end

Примеры использования цикла for (рис. [**fig:002?**]):

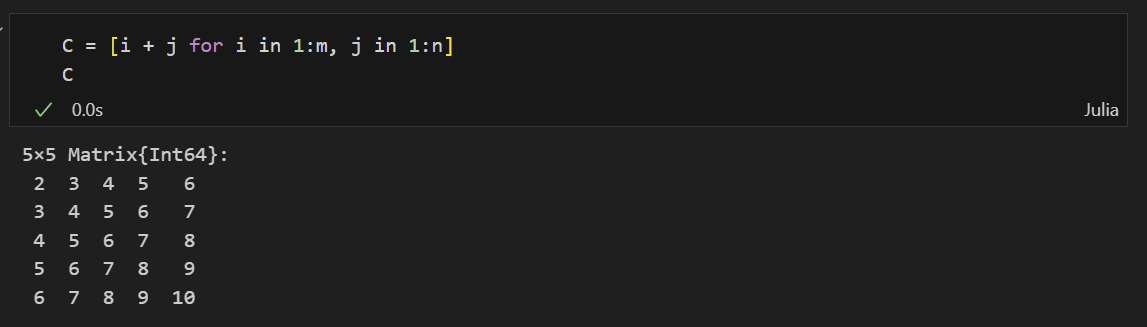


Примеры использования цикла for

Пример использования цикла for для создания двумерного массива, в котором значение каждой записи является суммой индексов строки и столбца (рис. [**fig:003?**] - рис. [**fig:004?**]):



Пример использования цикла for для создания двумерного массива



Пример использования цикла for для создания двумерного массива

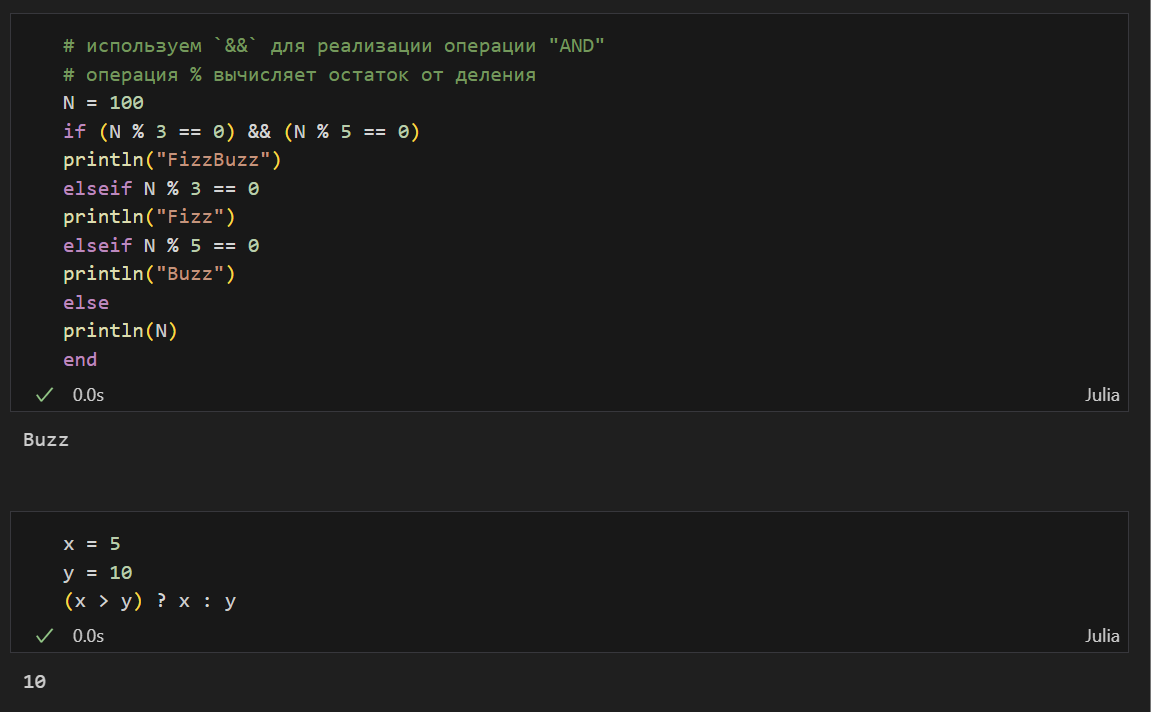
## 2.2 Условные выражения

Довольно часто при решении задач требуется проверить выполнение тех или иных условий. Для этого используют условные выражения.

Синтаксис условных выражений с ключевым словом:

if <condition 1>  
 <action 1>  
elseif <condition 2>  
 <action 2>  
else  
 <action 3>  
end

Примеры использования условного выражения (рис. [**fig:005?**]):

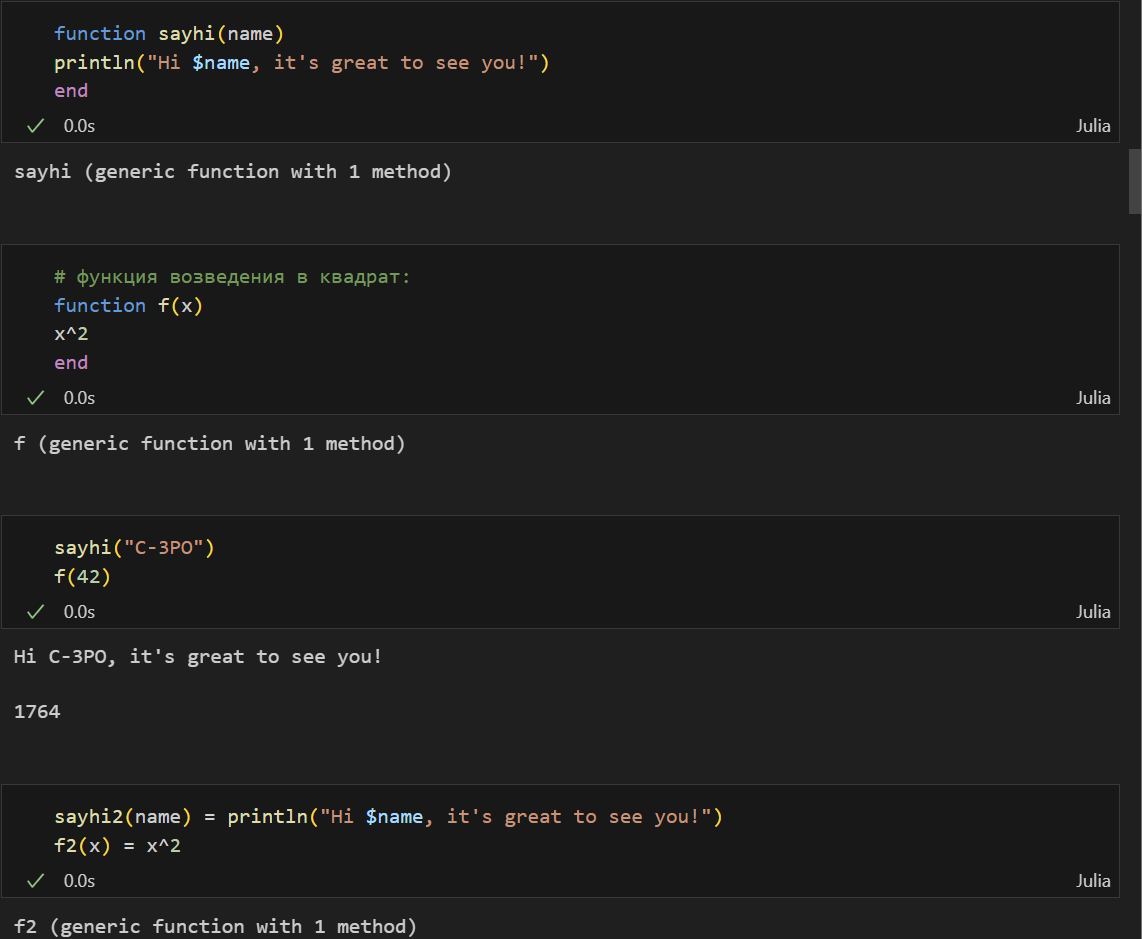


Примеры использования условного выражения

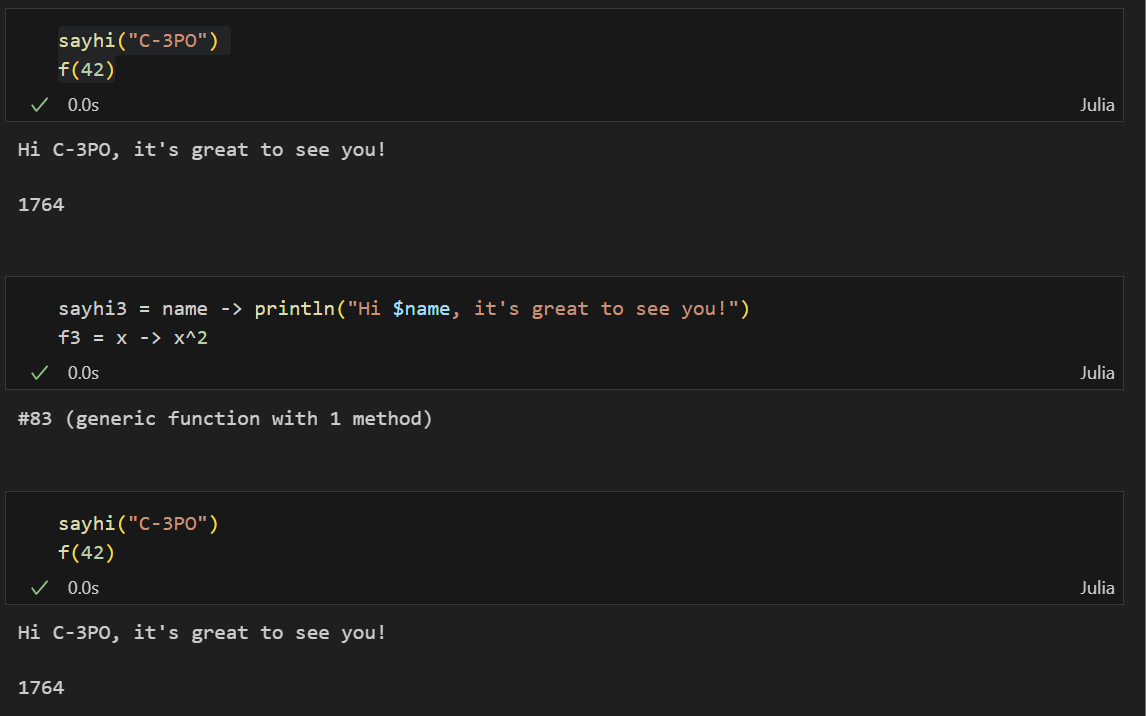
## 2.3 Функции

Julia дает нам несколько разных способов написать функцию.

Примеры способов написания функции (рис. [**fig:006?**] - рис. [**fig:007?**]):



Примеры способов написания функции



Примеры способов написания функции

По соглашению в Julia функции, сопровождаемые восклицательным знаком, изменяют свое содержимое, а функции без восклицательного знака не делают этого (рис. [**fig:008?**]):



Сравнение результатов вывода

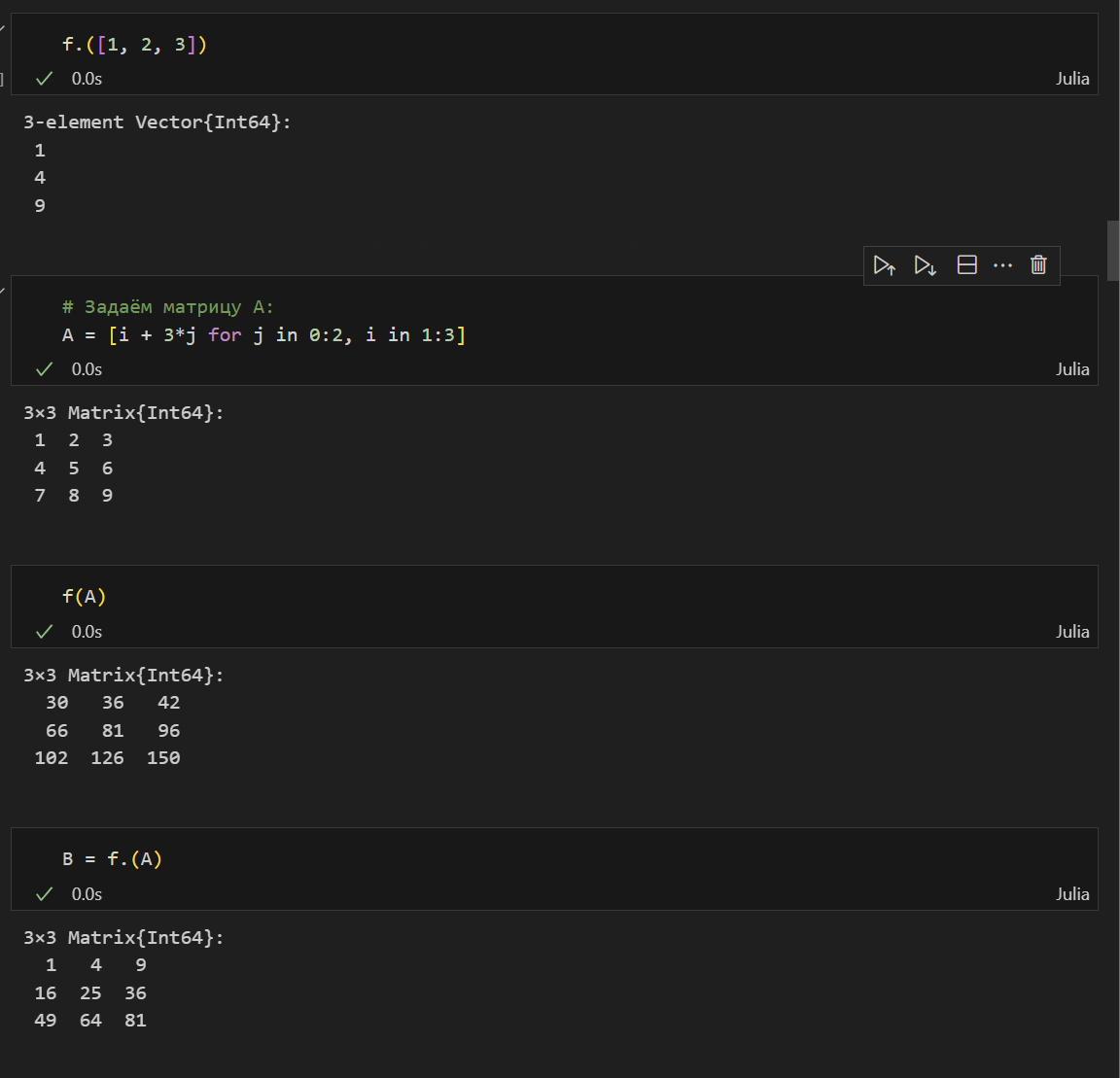
В Julia функция map является функцией высшего порядка, которая принимает функцию в качестве одного из своих входных аргументов и применяет эту функцию к каждому элементу структуры данных, которая ей передаётся также в качестве аргумента.

Функция broadcast — ещё одна функция высшего порядка в Julia, представляющая собой обобщение функции map.Функция broadcast() будет пытаться привести все объекты к общему измерению, map() будет напрямую применять данную функцию поэлементно.

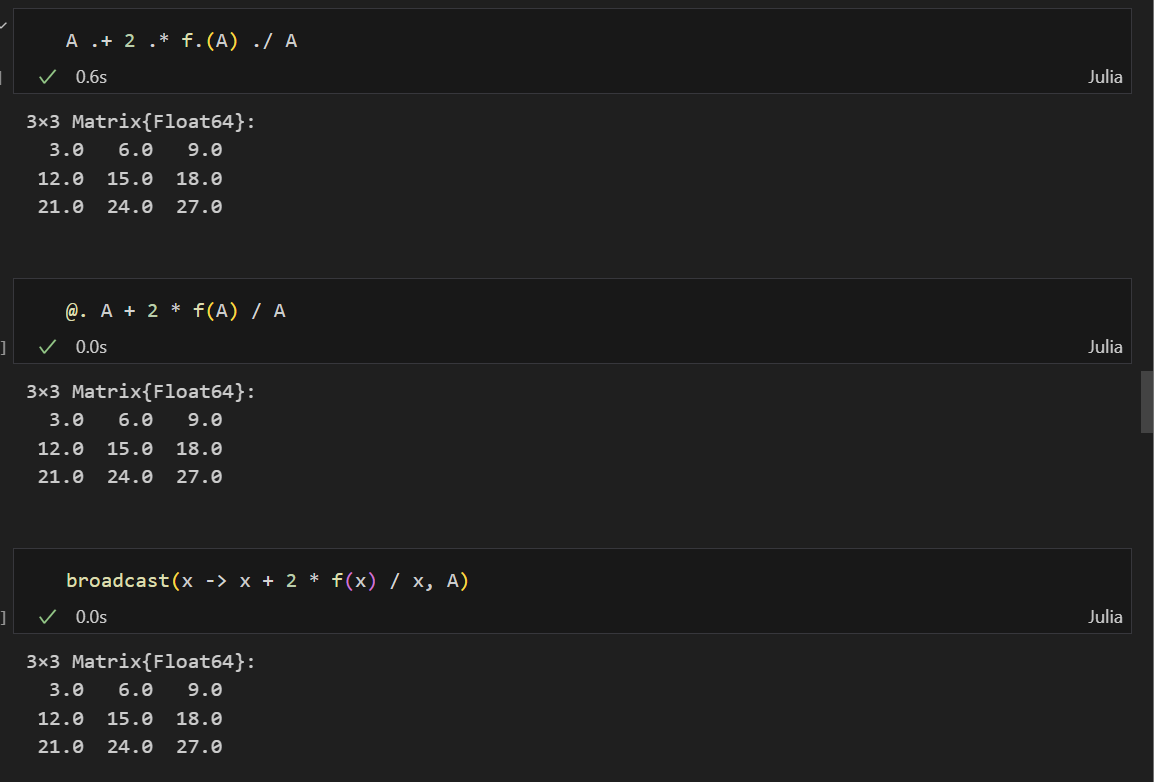
Примеры использования функций map() и broadcast() (рис. [**fig:009?**] - рис. [**fig:011?**]):



Примеры использования функций map() и broadcast()



Примеры использования функций map() и broadcast()



Примеры использования функций map() и broadcast()

## 2.4 Сторонние библиотеки (пакеты) в Julia

Julia имеет более 2000 зарегистрированных пакетов, что делает их огромной частью экосистемы Julia. Есть вызовы функций первого класса для других языков, обеспечивающие интерфейсы сторонних функций. Можно вызвать функции из Python или R, например, с помощью PyCall или Rcall.

С перечнем доступных в Julia пакетов можно ознакомиться на страницах следующих ресурсов: - https://julialang.org/packages/ - https://juliahub.com/ui/Home - https://juliaobserver.com/ - https://github.com/svaksha/Julia.jl

При первом использовании пакета в вашей текущей установке Julia вам необходимо использовать менеджер пакетов, чтобы явно его добавить:

import Pkg  
Pkg.add("Example")

При каждом новом использовании Julia (например, в начале нового сеанса в REPL или открытии блокнота в первый раз) нужно загрузить пакет, используя ключевое слово using:

Например, добавим и загрузим пакет Colors:

Pkg.add("Colors")  
using Colors

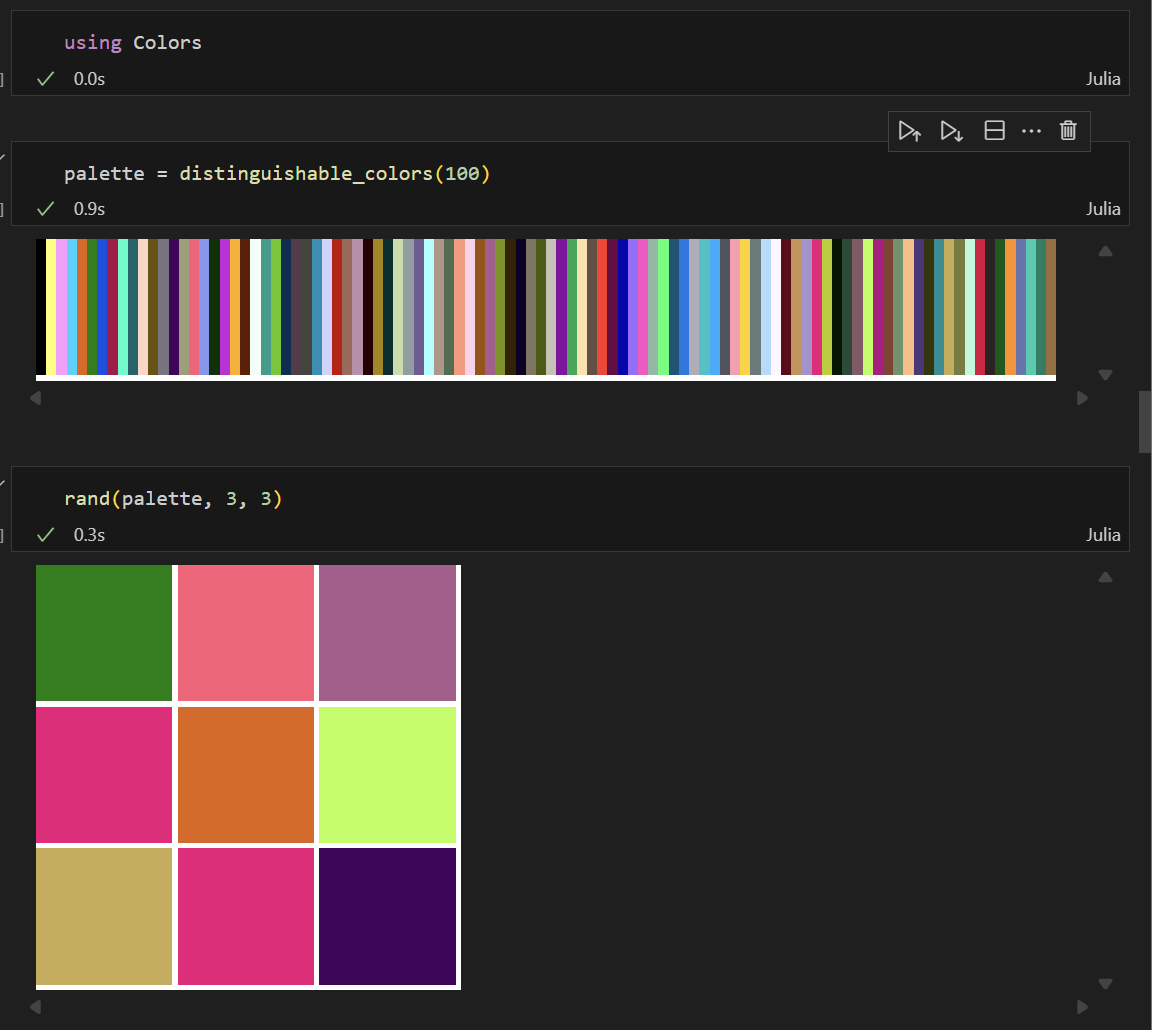
Затем создадим палитру из 100 разных цветов:

palette = distinguishable\_colors(100)

А затем определим матрицу 3 × 3 с элементами в форме случайного цвета из палитры, используя функцию rand:

rand(palette, 3, 3)

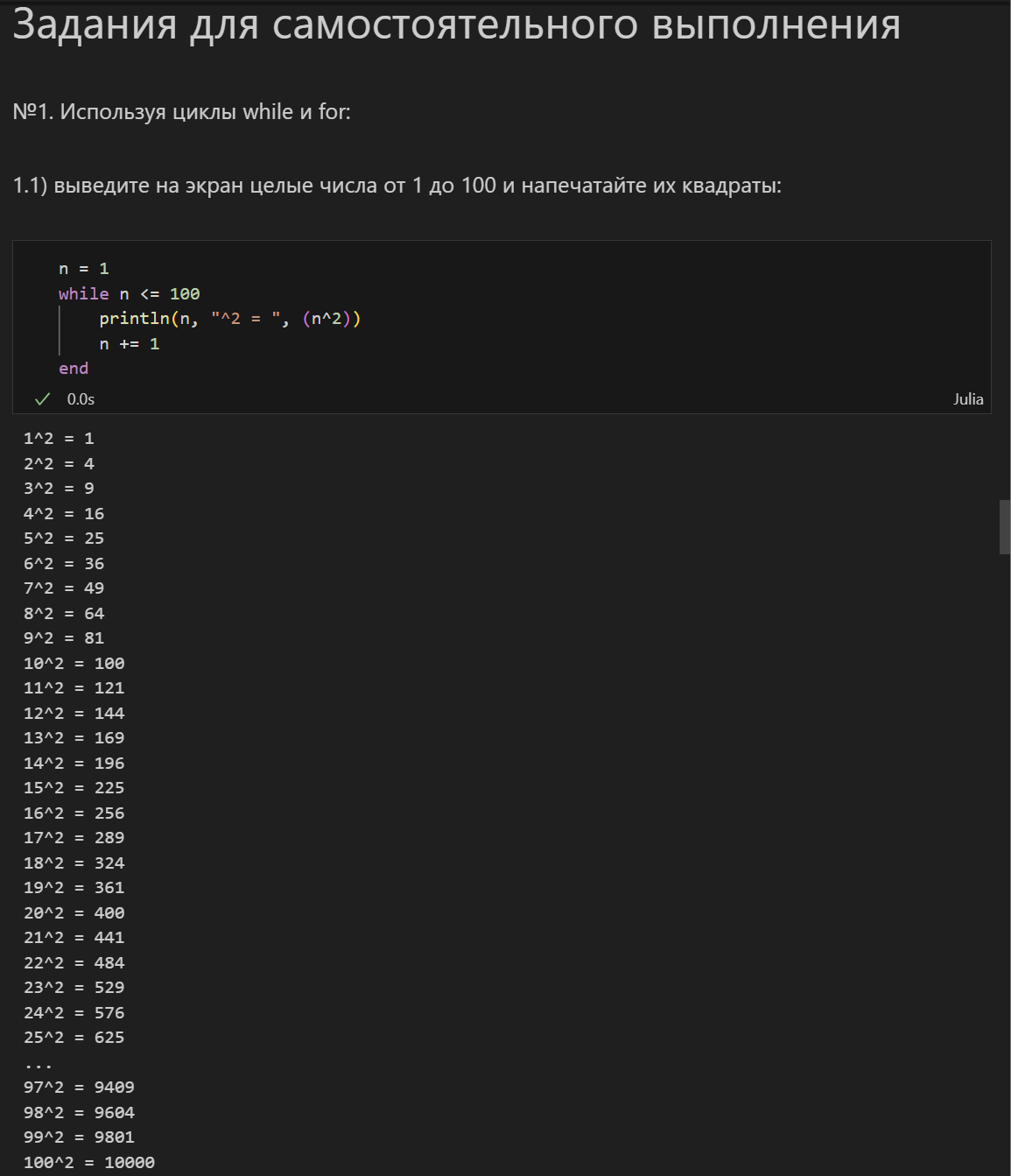
Пример использования сторонних библиотек (рис. [**fig:012?**]):



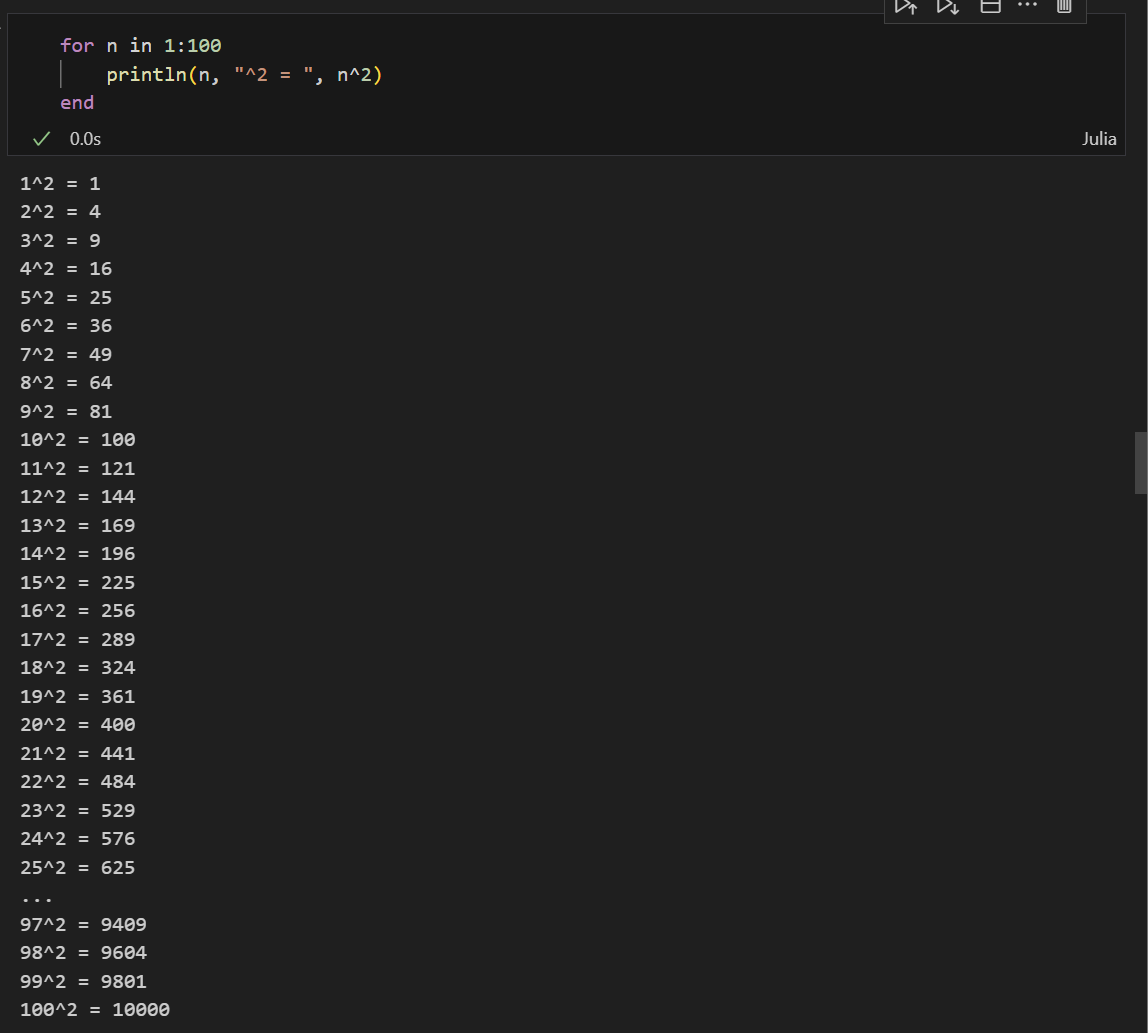
Пример использования сторонних библиотек

## 2.5 Самостоятельная работа

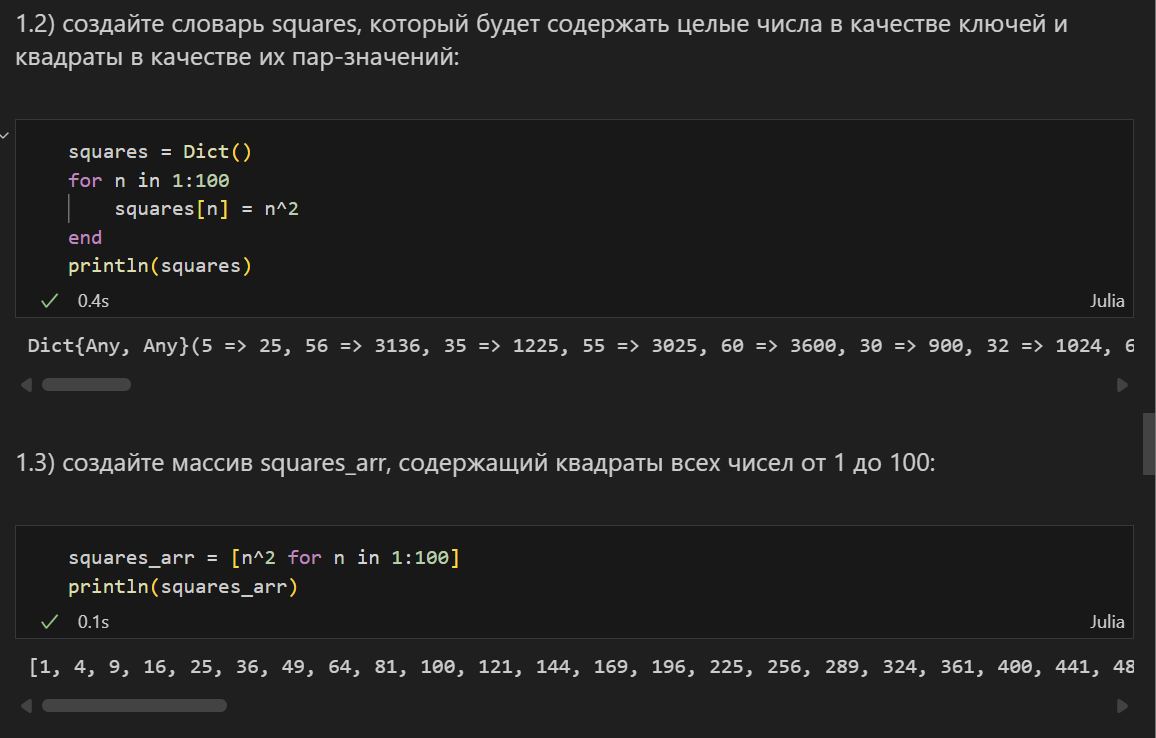
Выполнение задания №1 (рис. [**fig:013?**] - рис. [**fig:015?**]):



Выполнение подпунктов задания №1

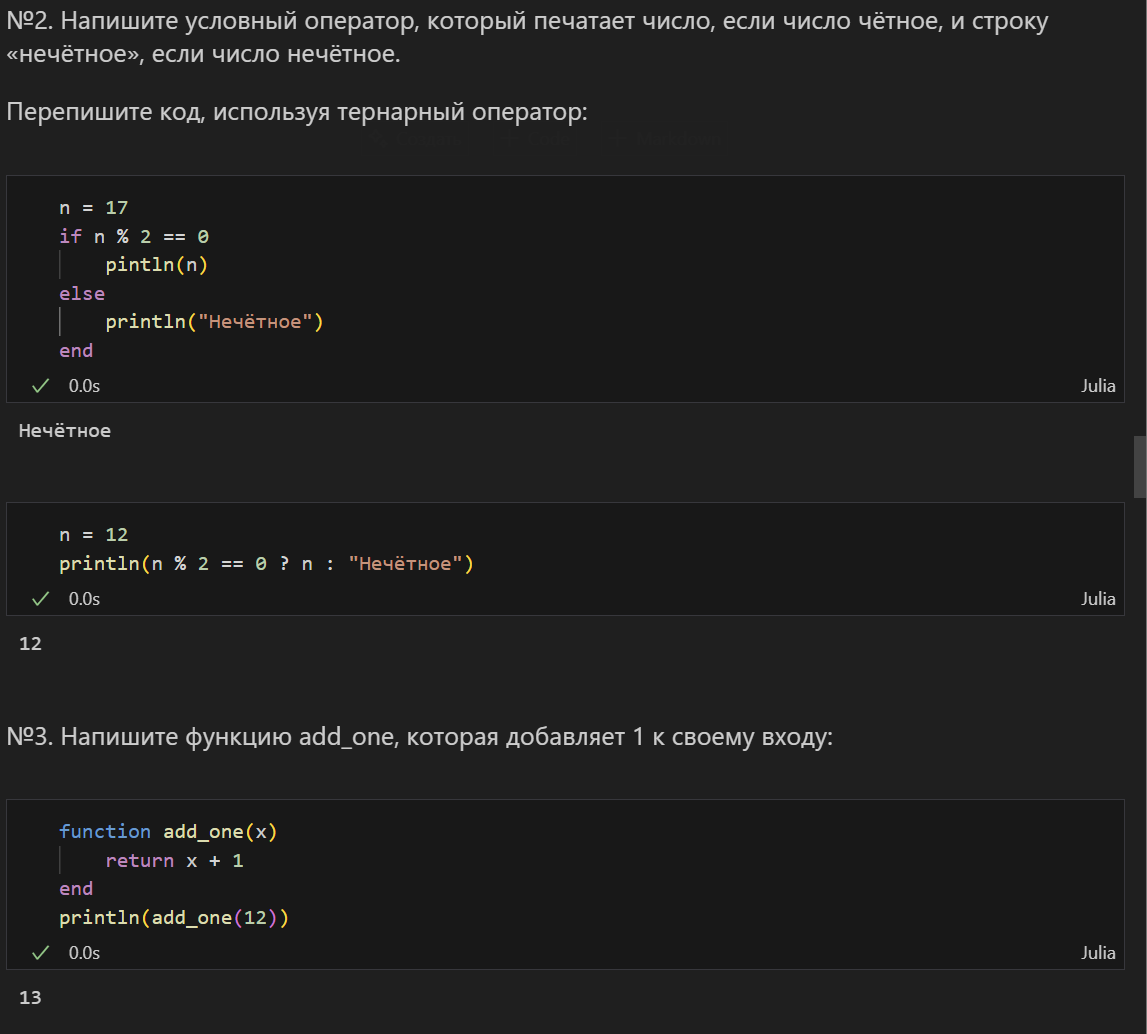


Выполнение подпунктов задания №1



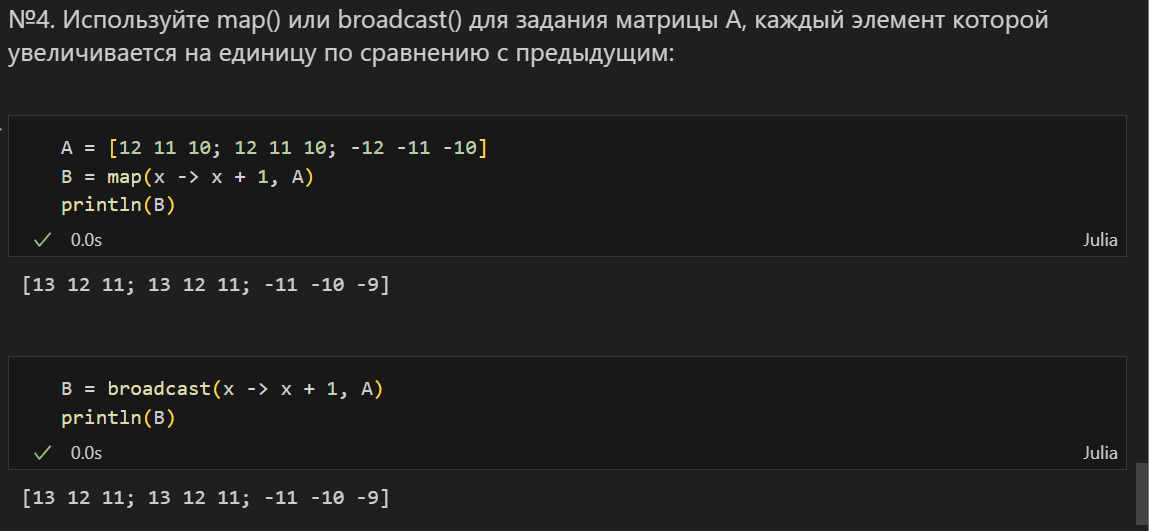
Выполнение подпунктов задания №1

Выполнение задания №2 и №3 (рис. [**fig:016?**]):



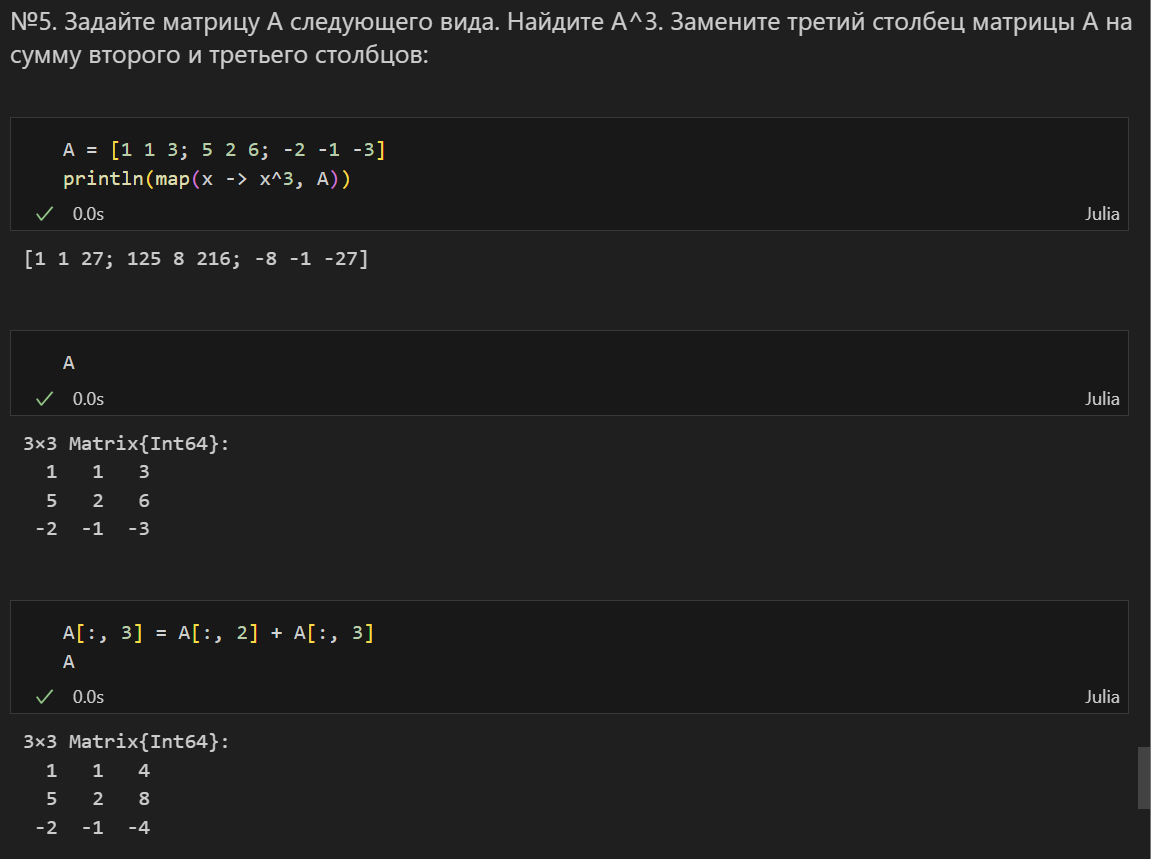
Выполнение задания №2 и №3

Выполнение задания №4 (рис. [**fig:017?**]):



Выполнение задания №4

Выполнение задания №5 (рис. [**fig:018?**]):



Выполнение задания №5

Выполнение задания №6 (рис. [**fig:019?**]):

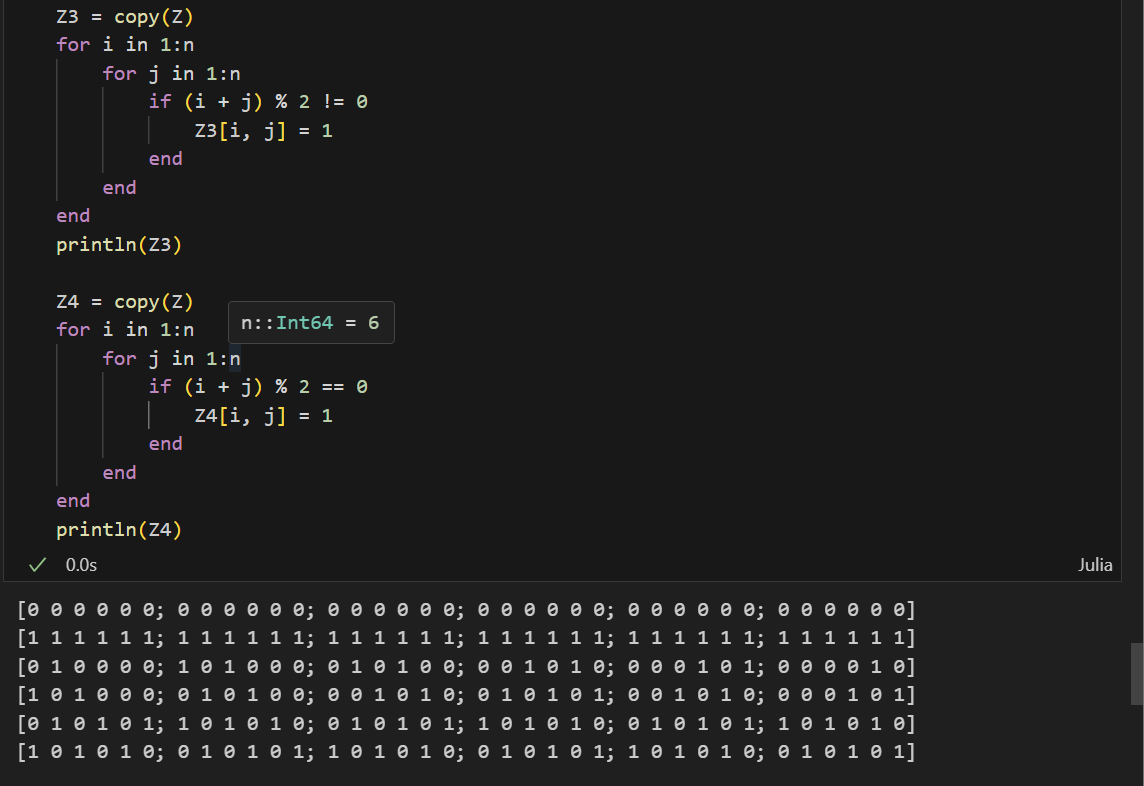


Выполнение задания №6

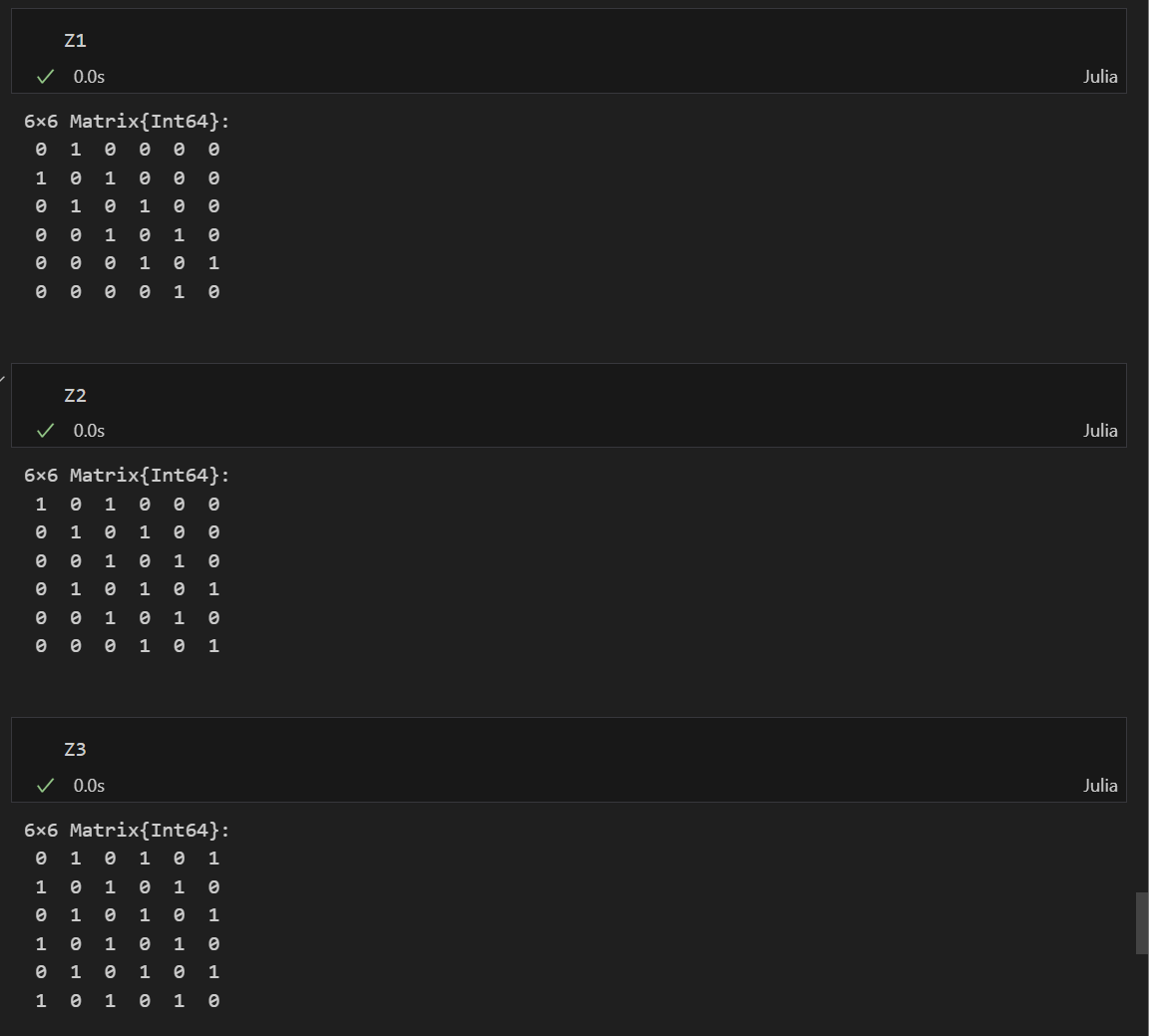
Выполнение задания №7 (рис. [**fig:020?**] - рис. [**fig:023?**]):



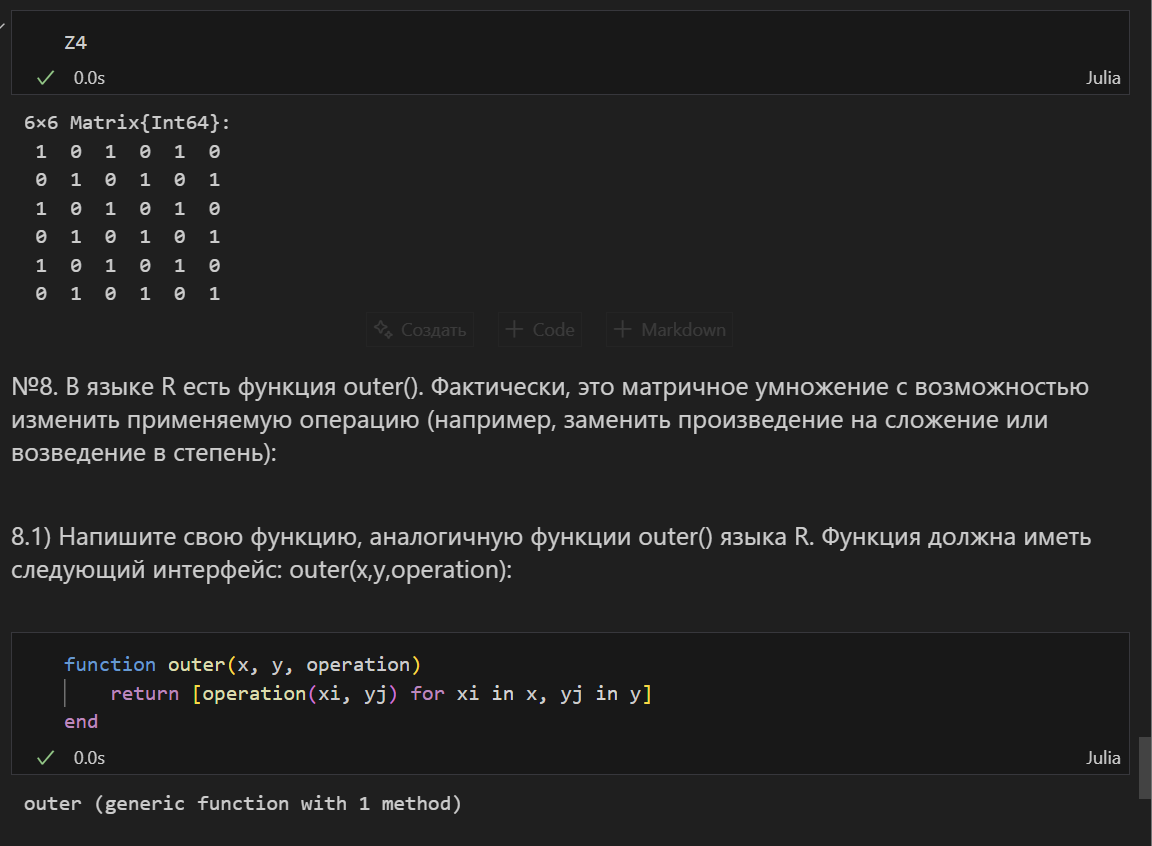
Выполнение задания №7



Выполнение задания №7

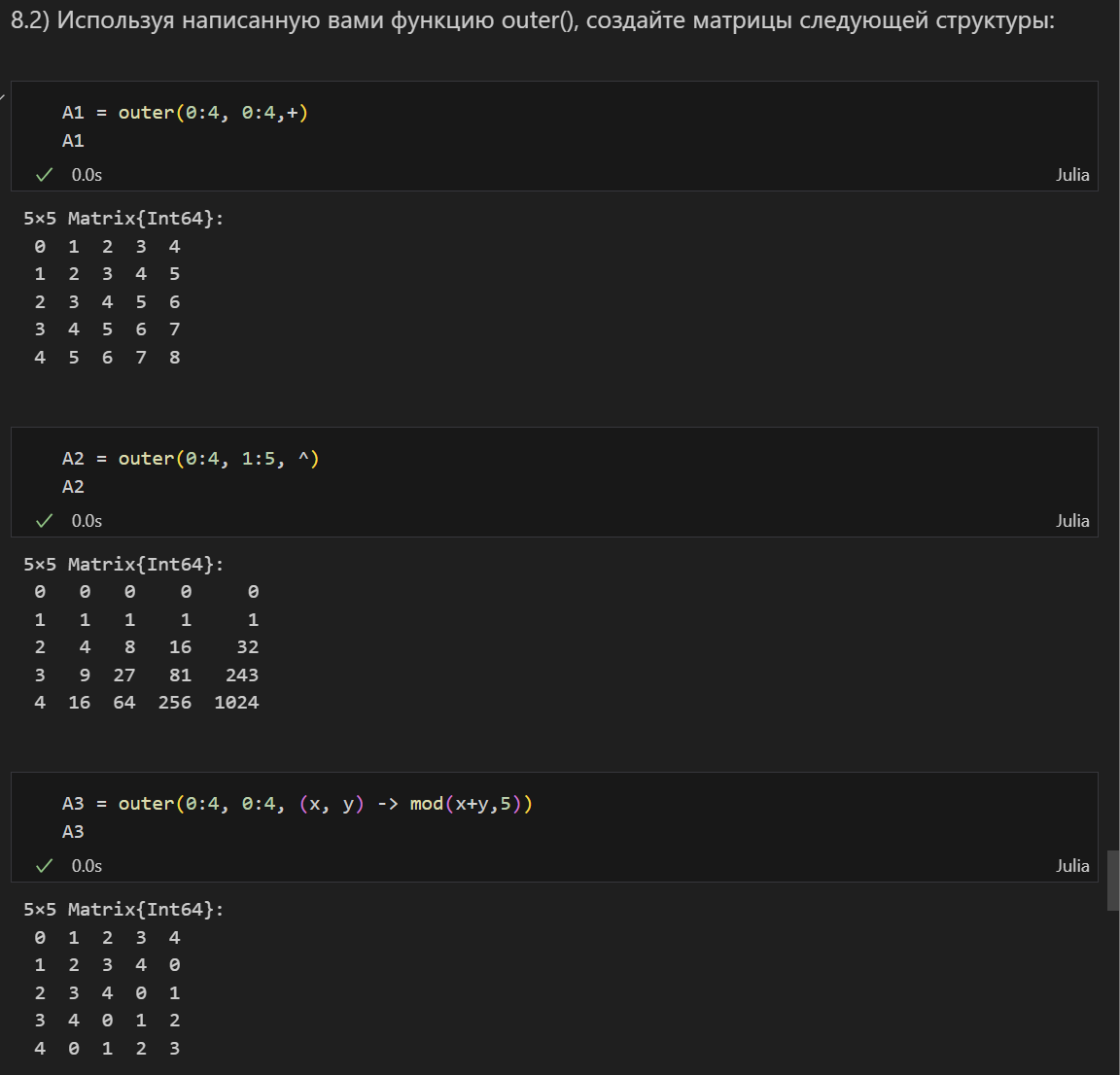


Выполнение задания №7

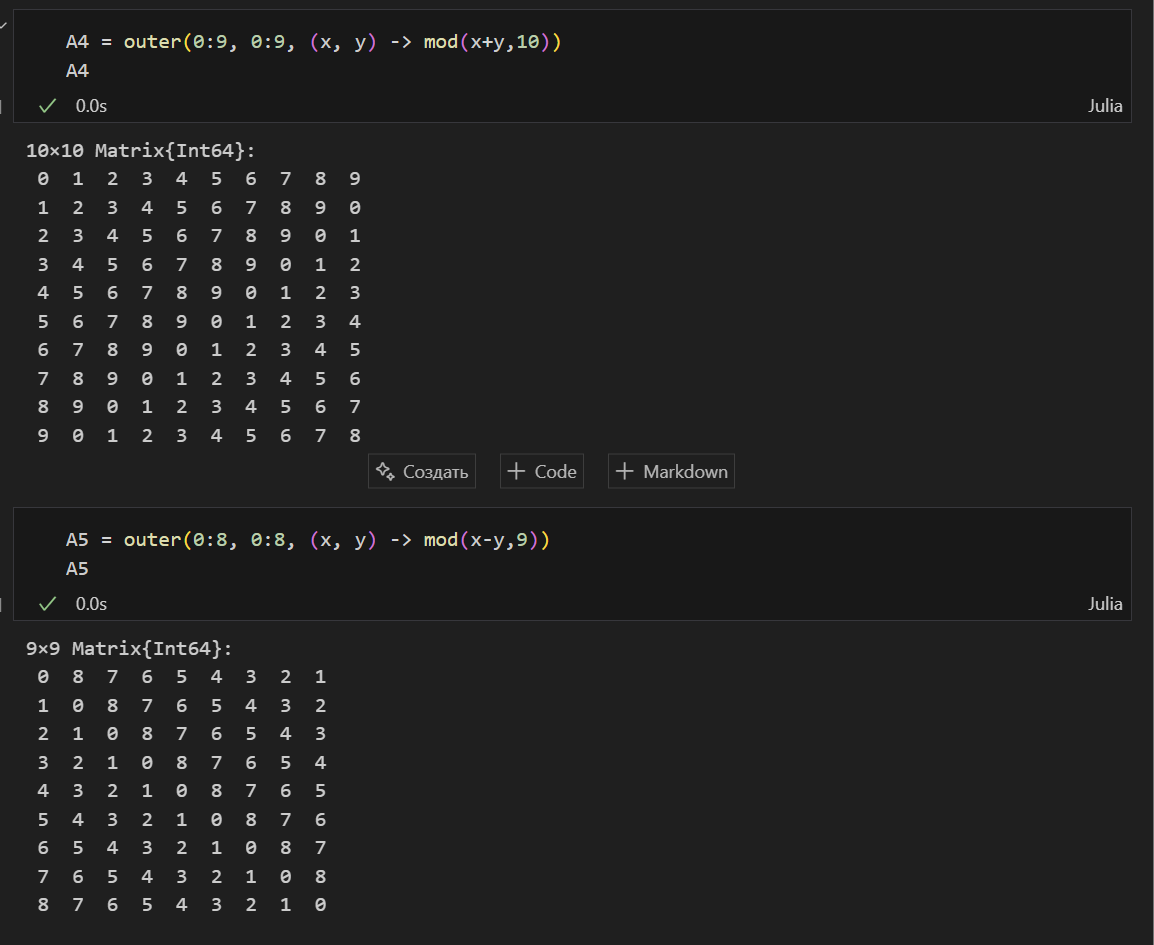


Выполнение задания №7 и №8

Выполнение задания №8 (рис. [**fig:024?**] - рис. [**fig:025?**]):

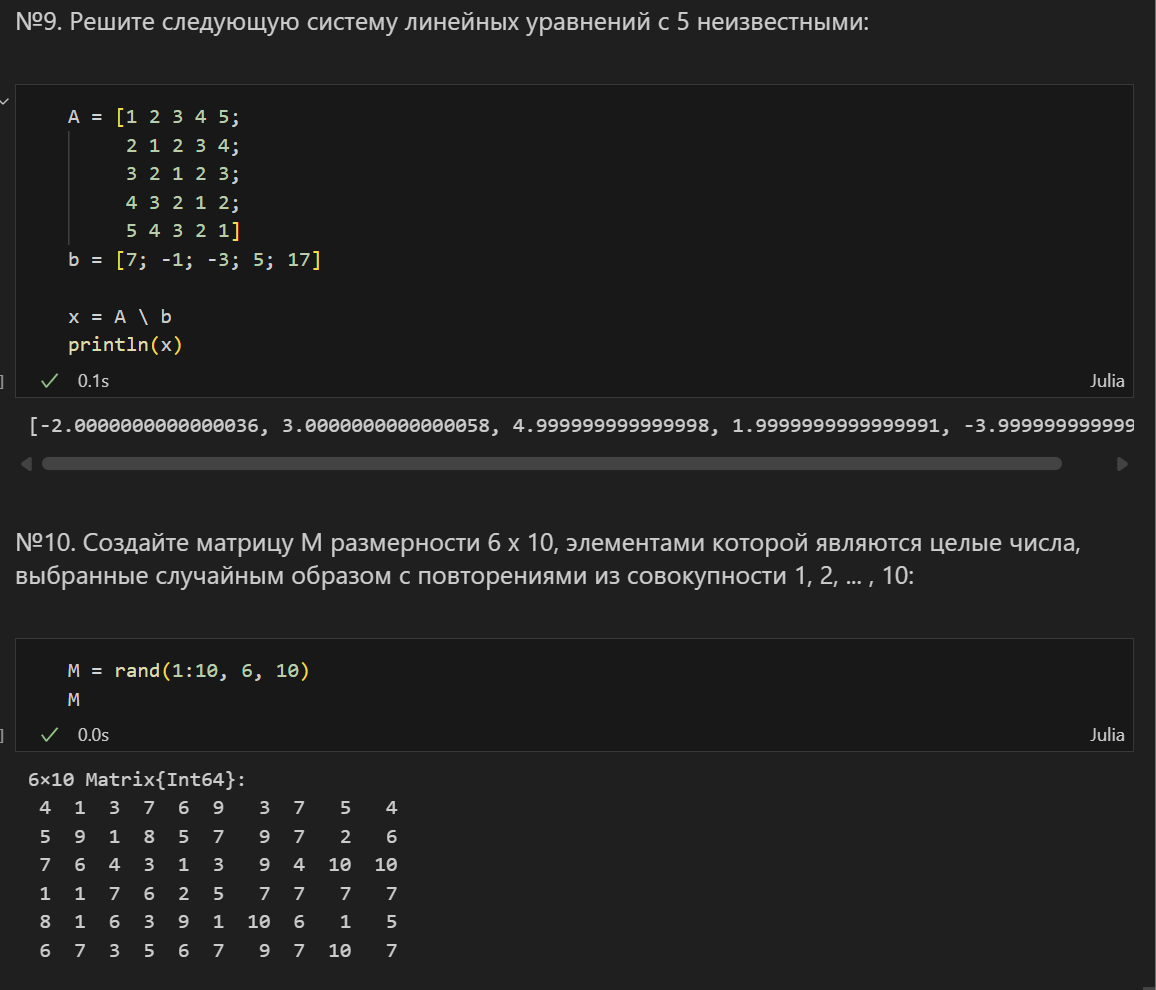


Выполнение подпунктов задания №8



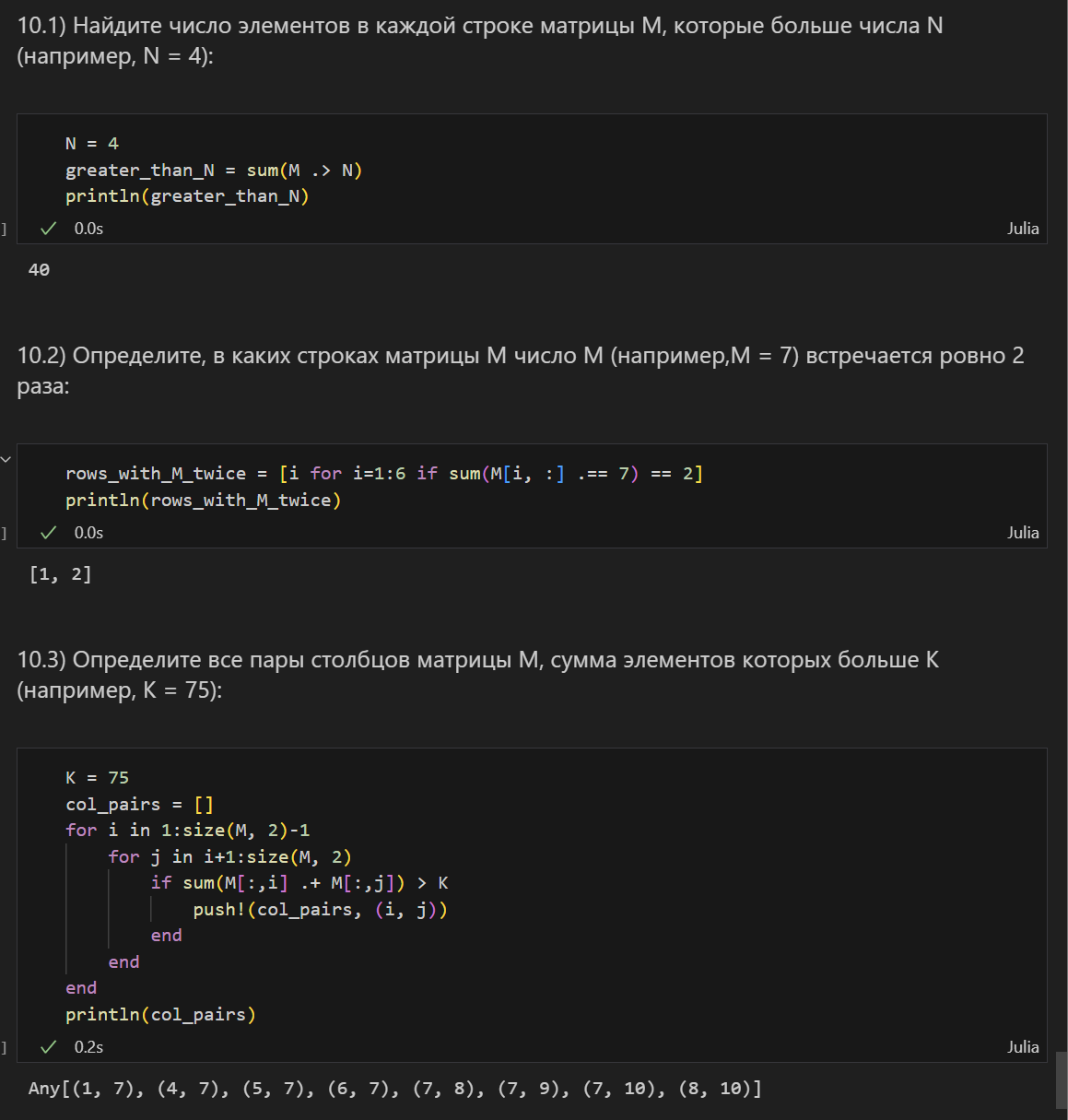
Выполнение подпунктов задания №8

Выполнение задания №9 и №10(рис. [**fig:026?**]):



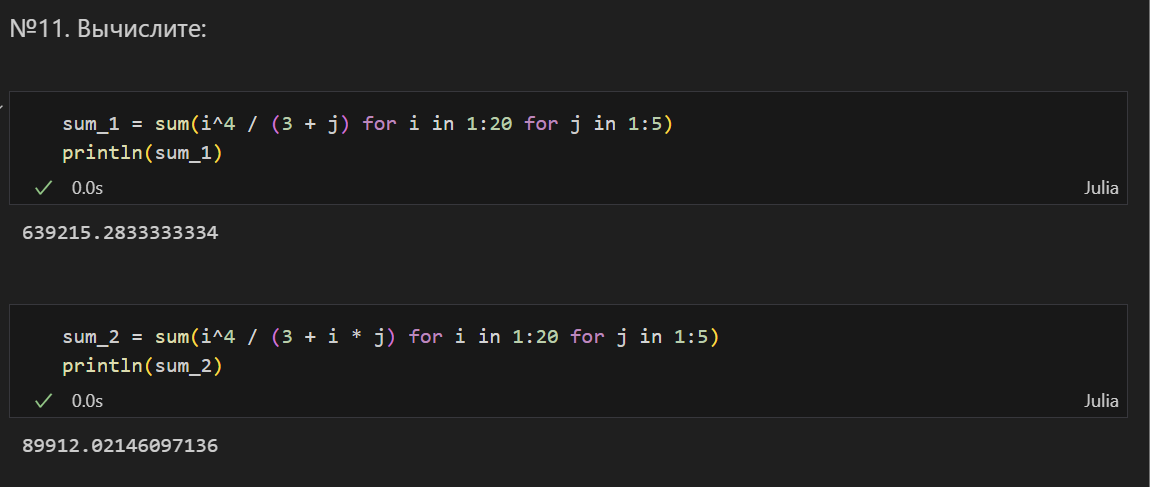
Выполнение задания №9 и №10

Выполнение задания №10 (рис. [**fig:027?**]):



Выполнение подпунктов задания №10

Выполнение задания №11 (рис. [**fig:028?**]):



Выполнение задания №11

# 3. Выводы

В результате выполнения данной лабораторной работы я освоил применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

# Список литературы

1. JuliaLang [Электронный ресурс]. 2025 JuliaLang.org contributors. URL: <https://julialang.org/> (дата обращения: 09.13.2025).

2. Julia 1.11 Documentation [Электронный ресурс]. 2025 JuliaLang.org contributors. URL: <https://docs.julialang.org/en/v1/> (дата обращения: 09.13.2025).