Отчёт по лабораторной работе №3  
Управляющие структуры

Статический анализ данных

Выполнила: Коняева Марина Александровна,  
НФИбд-01-21, 1032217044

Содержание

# Цели лабораторной работы

Освоить применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

# Теоретическое введение

Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения.[1]

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

Такие же результаты можно получить при использовании цикла for. Синтаксис for for in end

Довольно часто при решении задач требуется проверить выполнение тех или иных условий. Для этого используют условные выражения. Синтаксис условных выражений с ключевым словом: if <условие 1> <действие 1> elseif <условие 2> <действие 2> else <действие 3> end

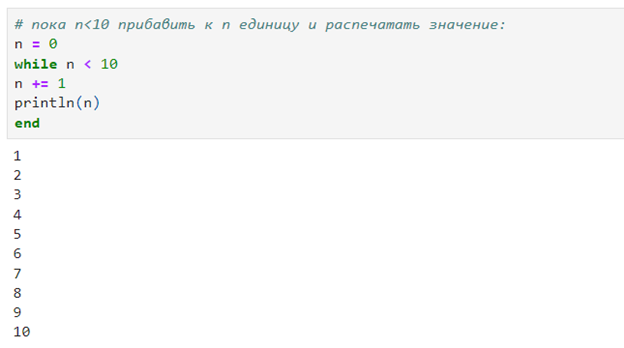
# Задачи лабораторной работы

1. Используя Jupyter Lab, повторите примеры из раздела 3.2.
2. Выполните задания для самостоятельной работы (раздел 3.4).

# Выполнение лабораторной работы

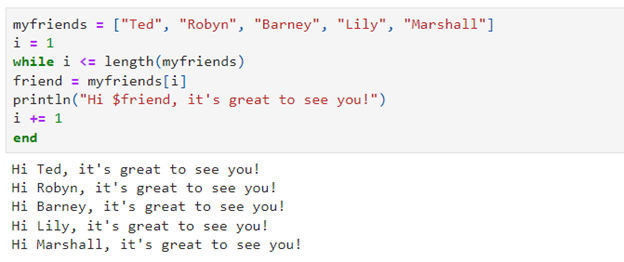
## Циклы while и for

1. Изучим информацию о циклах while и for, а также их синтаксис.
2. Повторим примеры с циклом while, а именно использование цикла для формирования элементов массива.



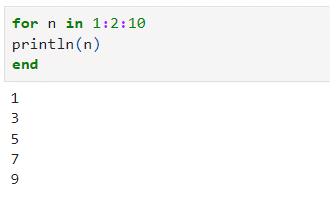
Формирования элементов массива

1. Повторим примеры с циклом while, а именно при работе со строковыми элементами массива, подставляя имя из массива в заданную строку приветствия и выводя получившуюся конструкцию на экран.



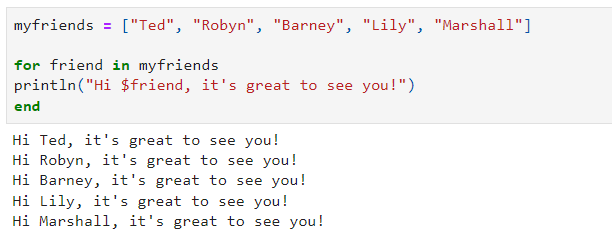
while при работе со строковыми элементами массива

1. Повторим примеры с циклом for, а именно использование цикла для формирования элементов массива.



Формирования элементов массива

1. Повторим примеры с циклом for, а именно при работе со строковыми элементами массива, подставляя имя из массива в заданную строку приветствия и выводя получившуюся конструкцию на экран.



for при работе со строковыми элементами массива

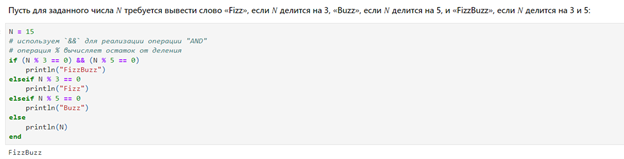
1. Рассмотрим три примера использования цикла for для создания двумерного массива, в котором значение каждой записи является суммой индексов строки и столбца.



Создания двумерного массива

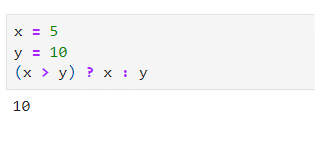
## Условные выражения

1. Изучим информацию об условных выражениях, а также их синтаксис. Синтаксис условных выражений с ключевым словом: if <условие 1> <действие 1> elseif <условие 2> <действие 2> else <действие 3> end
2. Повторим примеры с условными выражениями, а именно пусть для заданного числа 𝑁 требуется вывести слово «Fizz», если 𝑁 делится на 3, «Buzz», если 𝑁 делится на 5, и «FizzBuzz», если 𝑁 делится на 3 и 5.



Примеры с условными выражениями

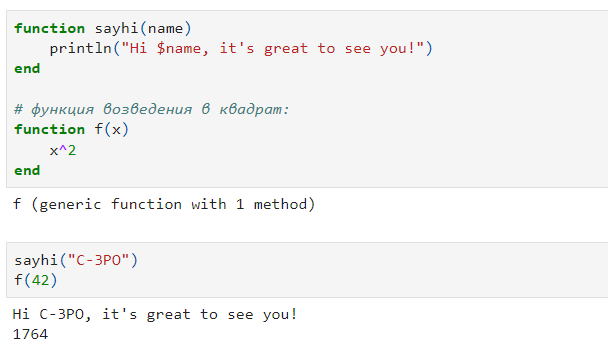
1. Повторим примеры с условными выражениями с тернарными операторами. Синтаксис условных выражений с тернарными операторами: a ? b : c (если выполнено a, то выполнить b, если нет, то c). Такая запись эквивалентна записи условного выражения с ключевым словом: if a b else c end



Пример с условными выражениями с тернарными операторами

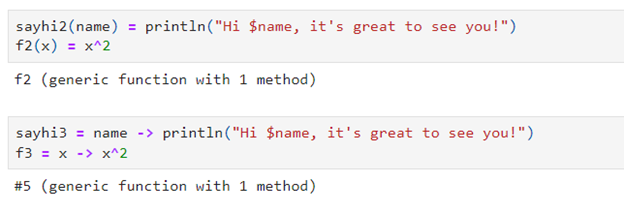
## Функции

1. Изучим информацию о функция и повторим примеры их задания, использования и вызов.



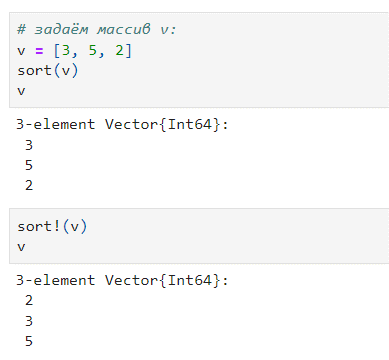
Функции (задание и вызов) 1

1. Повторим примеры с функциями, задание, использования и вызов, повторим альтернативу, можно объявить любую из выше определённых функций в одной строке, а также выполним пример, где можно объявить выше определённые функции как «анонимные».



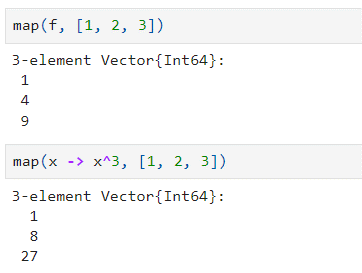
Функции (задание и вызов) 2-3

1. Выполним примеры с sort и sort!, по соглашению в Julia функции, сопровождаемые восклицательным знаком, изменяют свое содержимое, а функции без восклицательного знака не делают этого. Функция sort(v) возвращает отсортированный массив, который содержит те же элементы, что и массив v, но исходный массив v остаётся без изменений. Если же использовать sort!(v), то отсортировано будет содержимое исходного массива v.



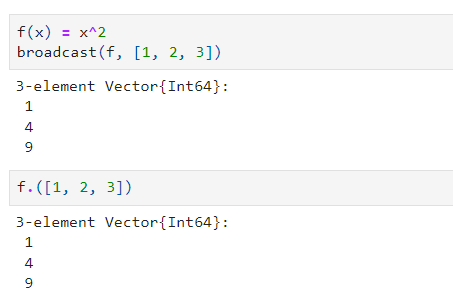
Примеры с sort и sort!

1. Повторим примеры с функцией map, 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.



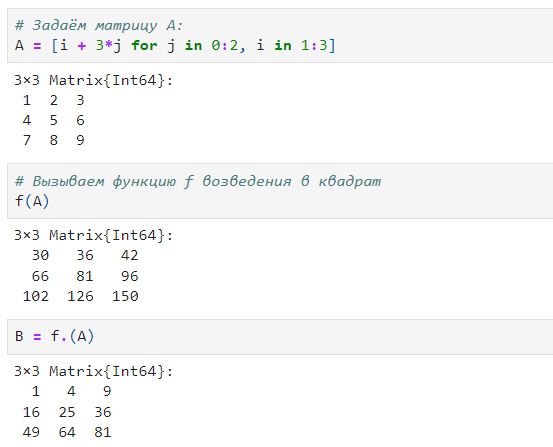
Примеры с map

1. Повторим примеры с функцией maр, в map можно передать и анонимно заданную, а не именованную функцию.



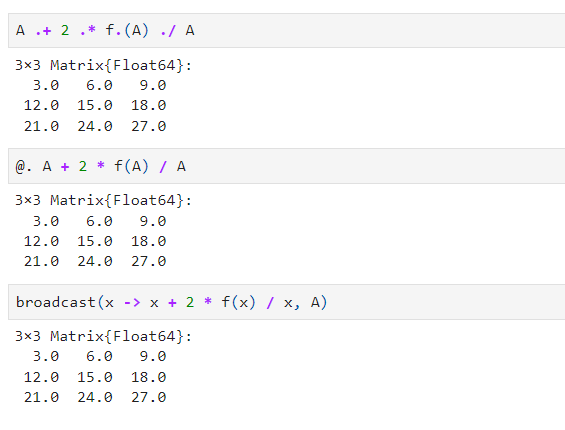
Примеры с map

1. Выполним примеры с функцией broadcast, 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. Синтаксис для вызова broadcast такой же, как и для вызоваmap, например применение функции f к элементам массива [1, 2, 3]: f(x) = x^2 broadcast(f, [1, 2, 3]) или f.([1, 2, 3])



Примеры с broadcast

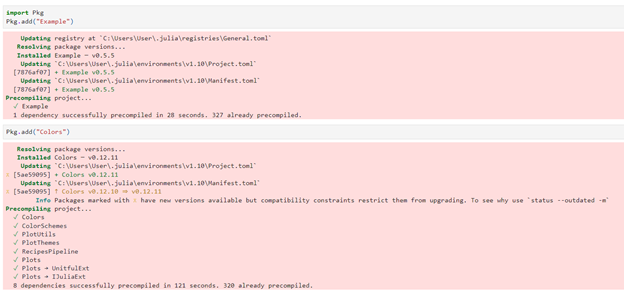
1. Выполним примеры с функцией broadcast, точечный синтаксис для broadcast() позволяет записать относительно сложные составные поэлементные выражения в форме, близкой к математической записи.



Примеры с broadcast

## Сторонние библиотеки (пакеты) в Julia

1. Изучим информацию о сторонних библиотеках. 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
2. Добавим необходимые пакеты для дальнейшего использования.



Сторонние библиотеки (пакеты)

1. Повторим примеры со сторонними библиотеками (пакетами), создадим палитру из 100 разных цветов, а затем определим матрицу 3 × 3 с элементами в форме случайного цвета из палитры, используя функцию rand.

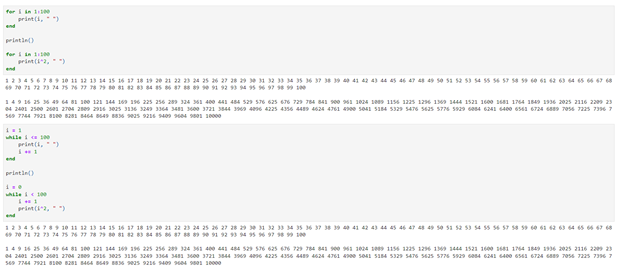


Пример со сторонними библиотеками

## Самостоятельная работа

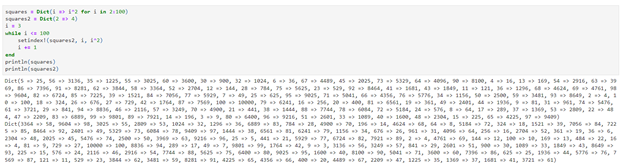
1. Выполним 1 задание для самостойтельной работы: Используя циклы while и for:

– выведите на экран целые числа от 1 до 100 и напечатайте их квадраты;

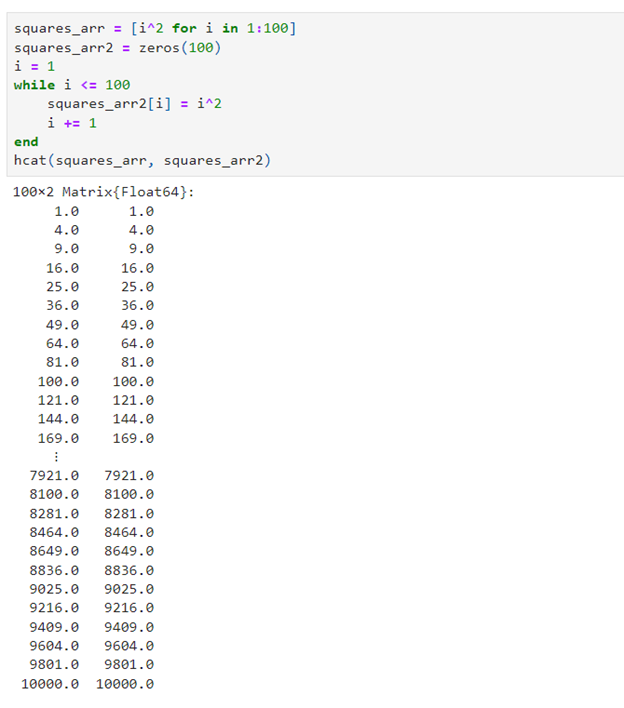


1 задания: часть 1

– создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;



1 задания: часть 2



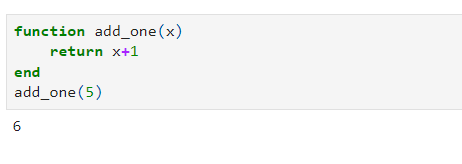
1 задания: часть 3

1. Выполним 2 задание для самостойтельной работы: напишите условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишите код, используя тернарный оператор.



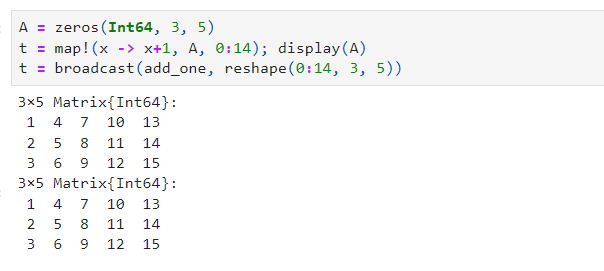
Выполнение 2 задания

1. Выполним 3 задание для самостойтельной работы: напишите функцию add\_one, которая добавляет 1 к своему входу.



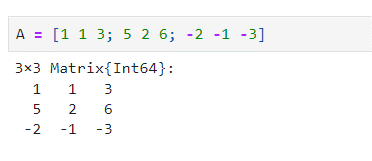
Выполнение 3 задания

1. Выполним 4 задание для самостойтельной работы: используйте map() или broadcast() для задания матрицы 𝐴, каждый элемент которой увеличивается на единицу по сравнению с предыдущим.



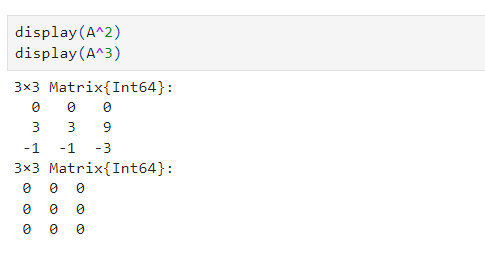
Задание 4

1. Выполним 5 задание для самостойтельной работы: Задайте матрицу следующего вида:



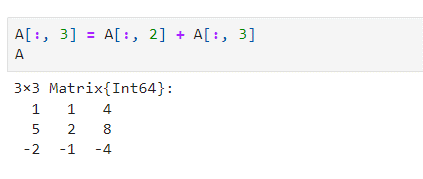
Задание 5

* Найдите



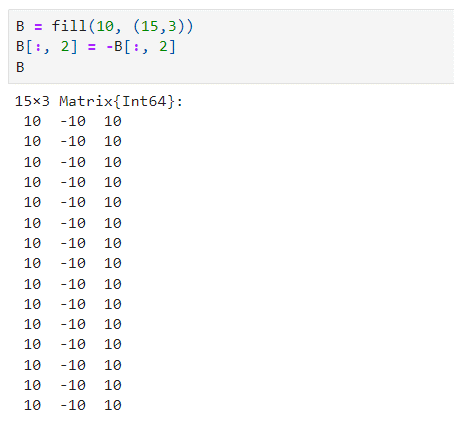
Задание 5: часть 1

* Замените третий столбец матрицы на сумму 2-го и 3-го столбцов



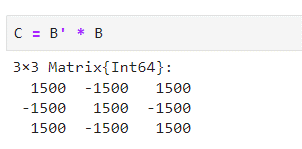
Задание 5: часть 2

1. Выполним 6 задание для самостойтельной работы: создайте матрицу с элементами .



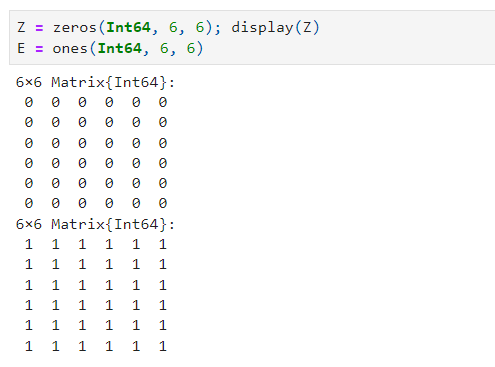
Задание 6

* Вычислите матрицу .



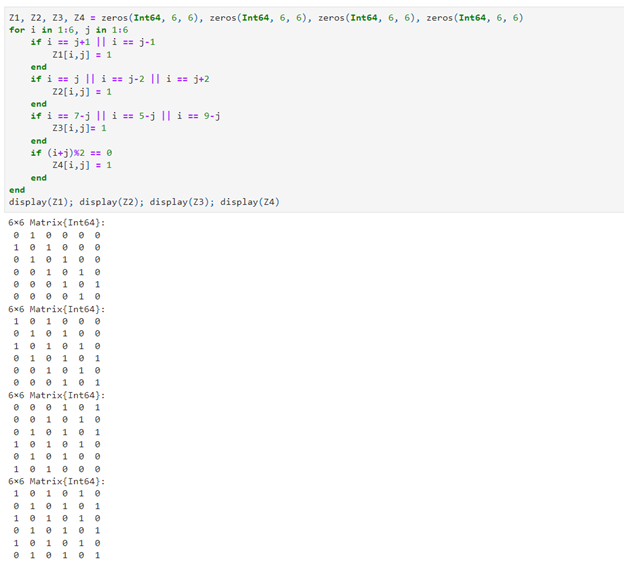
Задание 6

1. Выполним 7 задание для самостойтельной работы: создайте матрицу размерности , все элементы которой равны нулю, и матрицу , все элементы которой равны .



Задание 7

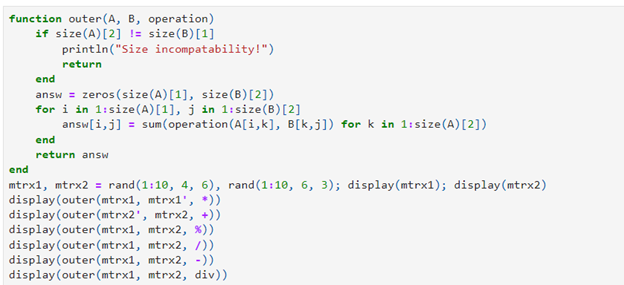
* Используя цикл while или for и закономерности расположения элементов, создайте следующие матрицы размерности :



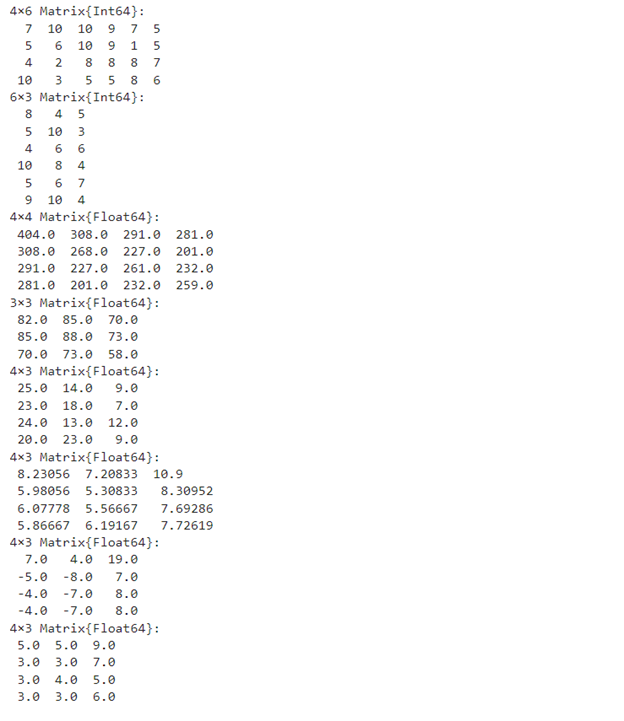
Задание 7: часть 1

1. Выполним 8 задание для самостойтельной работы: в языке R есть функция outer(). Фактически, это матричное умножение с возможностью изменить применяемую операцию (например, заменить произведение на сложение или возведение в степень).

* Напишите свою функцию, аналогичную функции outer() языка R. Функция должна иметь следующий интерфейс: outer(x,y,operation). Таким образом, функция вида outer(A,B,\*) должна быть эквивалентна произведению матриц и размерностями и соответственно, где элементы результирующей матрицы имеют вид (или в тензорном виде )



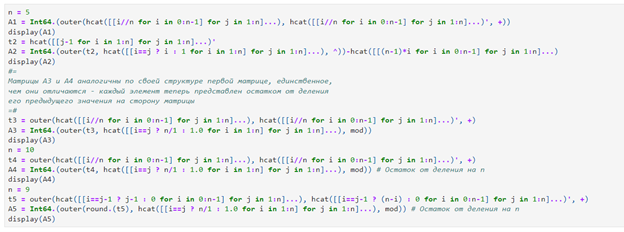
Задание 8: часть 1



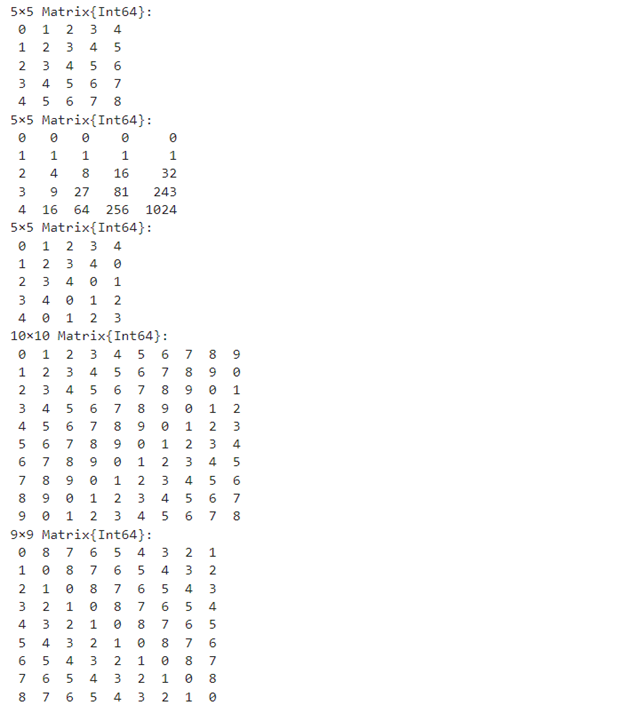
Задание 8: часть 1 (вывод)

* Используя написанную вами функцию outer(), создайте матрицы следующей структуры:

В каждом случае ваше решение должно быть легко обобщаемым на случай создания матриц большей размерности, но той же структуры.



Задание 8: часть 2

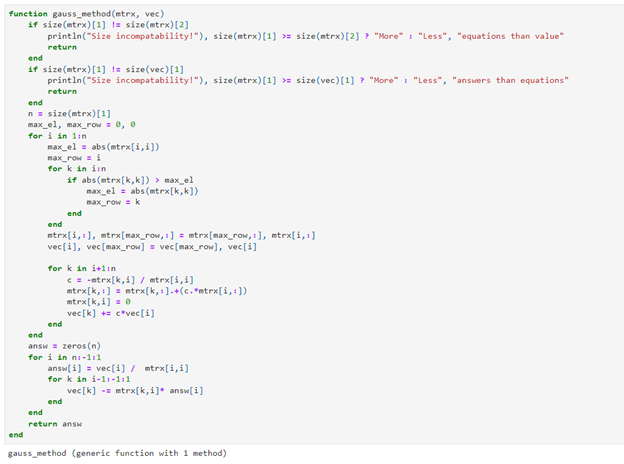


Задание 8: часть 2 (вывод)

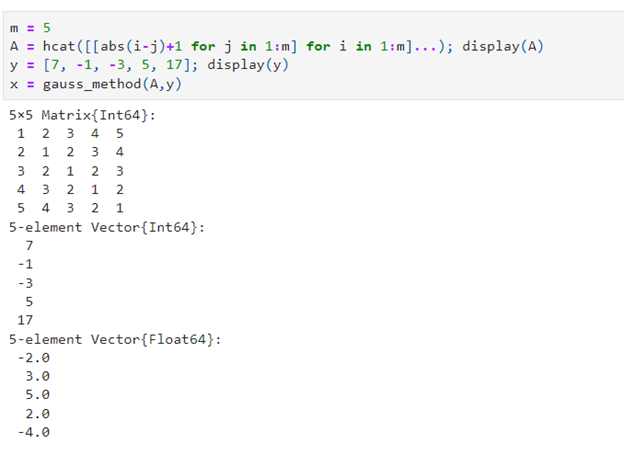
1. Выполним 9 задание для самостойтельной работы: решите следующую систему линейных уравнений с 5 неизвестными

рассмотрев соответствующее матричное уравнение . Обратите внимание на особый вид матрицы . Метод, используемый для решения данной системы уравнений, должен быть легко обобщаем на случай большего числа уравнений, где матрица будет иметь такую же структуру.

Решение будет осуществлено методом Гаусса.

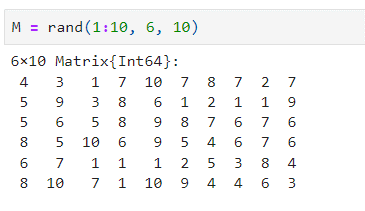


Задание 9: метод Гаусса



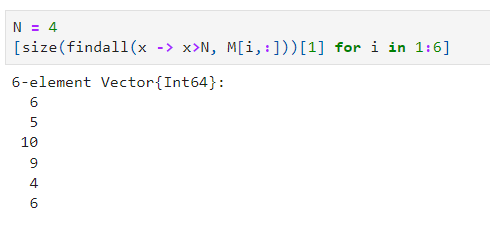
Задание 9: метод Гаусса (решение)

1. Выполним 10 задание для самостойтельной работы: создайте матрицу размерности , элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности .



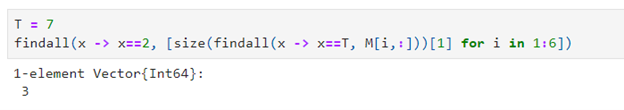
Задание 10

* Найдите число элементов в каждой строке матрицы , которые больше числа (например, ).



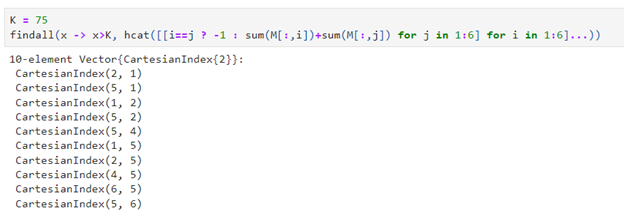
Задание 10: часть 1

* Определите, в каких строках матрицы число (например, ) встречается ровно 2 раза?



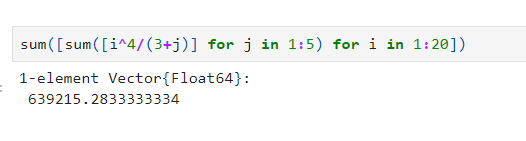
Задание 10: часть 2

* Определите все пары столбцов матрицы , сумма элементов которых больше (например, ).

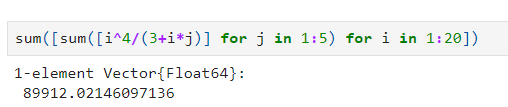


Задание 10: часть 3

1. Выполним 11 задание для самостойтельной работы: вычислите:



Задание 11: часть 1



Задание 11: часть 1

# Выводы по проделанной работе

## Вывод

В результате выполнения работы мы освоили применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

Были записаны скринкасты выполнения , создания отчета, презентации и защиты лабораторной работы.

# Список литературы

* Julia: https://ru.wikipedia.org/wiki/Julia
* https://julialang.org/packages/
* https://juliahub.com/ui/Home
* https://juliaobserver.com/
* https://github.com/svaksha/Julia.jl