РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

дисциплина: Компьютерный практикум по статистическому анализу данных

Студент: Евдокимов Иван Андреевич

Группа: НФИбд-01-20

МОСКВА

2023 г.

Цель работы:

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

In [1]:
```
;date
```

```
Fri Nov 24 15:54:13 RTZ 2023
```

In [2]:
```
;whoami
```

```
win-orl1fa978ms\admin
```

1. Используя циклы $while$ и for:
   – выведите на экран целые числа от 1 до 100 и напечатайте их квадраты;(1.1)
   – создайте словарь $squares$, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;(1.2)

In [3]:
```
# (1.1) Используем цикл while для вывода целых чисел от 1 до 100 и их квадратов
println("Целые числа и их квадраты (цикл while):")
i = 1
while i <= 25
    square = i^2
    println("Число:", i, " ","Квадрат:", i^2 ,"    ", "Число:", i + 25," ", "Ква
    i += 1
end
```

```
Целые числа и их квадраты (цикл while):
Число:1 Квадрат:1    Число:26 Квадрат:676    Число:51 Квадрат:2601    Число:76 Кв
адрат:5776
Число:2 Квадрат:4    Число:27 Квадрат:729    Число:52 Квадрат:2704    Число:77 Кв
адрат:5929
Число:3 Квадрат:9    Число:28 Квадрат:784    Число:53 Квадрат:2809    Число:78 Кв
адрат:6084
Число:4 Квадрат:16    Число:29 Квадрат:841    Число:54 Квадрат:2916    Число:79 К
вадрат:6241
Число:5 Квадрат:25    Число:30 Квадрат:900    Число:55 Квадрат:3025    Число:80 К
вадрат:6400
Число:6 Квадрат:36    Число:31 Квадрат:961    Число:56 Квадрат:3136    Число:81 К
вадрат:6561
Число:7 Квадрат:49    Число:32 Квадрат:1024    Число:57 Квадрат:3249    Число:82
Квадрат:6724
Число:8 Квадрат:64    Число:33 Квадрат:1089    Число:58 Квадрат:3364    Число:83
Квадрат:6889
Число:9 Квадрат:81    Число:34 Квадрат:1156    Число:59 Квадрат:3481    Число:84
Квадрат:7056
Число:10 Квадрат:100    Число:35 Квадрат:1225    Число:60 Квадрат:3600    Число:8
5 Квадрат:7225
Число:11 Квадрат:121    Число:36 Квадрат:1296    Число:61 Квадрат:3721    Число:8
6 Квадрат:7396
Число:12 Квадрат:144    Число:37 Квадрат:1369    Число:62 Квадрат:3844    Число:8
7 Квадрат:7569
Число:13 Квадрат:169    Число:38 Квадрат:1444    Число:63 Квадрат:3969    Число:8
8 Квадрат:7744
Число:14 Квадрат:196    Число:39 Квадрат:1521    Число:64 Квадрат:4096    Число:8
9 Квадрат:7921
Число:15 Квадрат:225    Число:40 Квадрат:1600    Число:65 Квадрат:4225    Число:9
0 Квадрат:8100
Число:16 Квадрат:256    Число:41 Квадрат:1681    Число:66 Квадрат:4356    Число:9
1 Квадрат:8281
Число:17 Квадрат:289    Число:42 Квадрат:1764    Число:67 Квадрат:4489    Число:9
2 Квадрат:8464
Число:18 Квадрат:324    Число:43 Квадрат:1849    Число:68 Квадрат:4624    Число:9
3 Квадрат:8649
Число:19 Квадрат:361    Число:44 Квадрат:1936    Число:69 Квадрат:4761    Число:9
4 Квадрат:8836
Число:20 Квадрат:400    Число:45 Квадрат:2025    Число:70 Квадрат:4900    Число:9
5 Квадрат:9025
Число:21 Квадрат:441    Число:46 Квадрат:2116    Число:71 Квадрат:5041    Число:9
6 Квадрат:9216
Число:22 Квадрат:484    Число:47 Квадрат:2209    Число:72 Квадрат:5184    Число:9
7 Квадрат:9409
Число:23 Квадрат:529    Число:48 Квадрат:2304    Число:73 Квадрат:5329    Число:9
8 Квадрат:9604
Число:24 Квадрат:576    Число:49 Квадрат:2401    Число:74 Квадрат:5476    Число:9
9 Квадрат:9801
Число:25 Квадрат:625    Число:50 Квадрат:2500    Число:75 Квадрат:5625    Число:1
00 Квадрат:10000
```

In [4]:

```
# (1.2) Используем цикл for для создания словаря squares
squares = Dict{Int, Int}()

println("\nСоздание словаря squares (цикл for):")
for j in 1:100
    squares[j] = j^2
end
```

```
println(squares)
```

```
Создание словаря squares (цикл for):
Dict(5 => 25, 56 => 3136, 35 => 1225, 55 => 3025, 60 => 3600, 30 => 900, 32 => 10
24, 6 => 36, 67 => 4489, 45 => 2025, 73 => 5329, 64 => 4096, 90 => 8100, 4 => 16,
13 => 169, 54 => 2916, 63 => 3969, 86 => 7396, 91 => 8281, 62 => 3844, 58 => 336
4, 52 => 2704, 12 => 144, 28 => 784, 75 => 5625, 23 => 529, 92 => 8464, 41 => 168
1, 43 => 1849, 11 => 121, 36 => 1296, 68 => 4624, 69 => 4761, 98 => 9604, 82 => 6
724, 85 => 7225, 39 => 1521, 84 => 7056, 77 => 5929, 7 => 49, 25 => 625, 95 => 90
25, 71 => 5041, 66 => 4356, 76 => 5776, 34 => 1156, 50 => 2500, 59 => 3481, 93 =>
8649, 2 => 4, 10 => 100, 18 => 324, 26 => 676, 27 => 729, 42 => 1764, 87 => 7569,
100 => 10000, 79 => 6241, 16 => 256, 20 => 400, 81 => 6561, 19 => 361, 49 => 240
1, 44 => 1936, 9 => 81, 31 => 961, 74 => 5476, 61 => 3721, 29 => 841, 94 => 8836,
46 => 2116, 57 => 3249, 70 => 4900, 21 => 441, 38 => 1444, 88 => 7744, 78 => 608
4, 72 => 5184, 24 => 576, 8 => 64, 17 => 289, 37 => 1369, 1 => 1, 53 => 2809, 22
=> 484, 47 => 2209, 83 => 6889, 99 => 9801, 89 => 7921, 14 => 196, 3 => 9, 80 =>
6400, 96 => 9216, 51 => 2601, 33 => 1089, 40 => 1600, 48 => 2304, 15 => 225, 65 =
> 4225, 97 => 9409)
```

2. Напишите условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное(2.1). Перепишите код, используя тернарный оператор(2.2).

In [5]:
```
#(2.1)
function printNumberType(num)
    if iseven(num)
        println("Чётное число: $num")
    else
        println("Нечётное число")
    end
end

# Пример использования
printNumberType(4)
printNumberType(7)
printNumberType(76)
```

```
Чётное число: 4
Нечётное число
Чётное число: 76
```

In [6]:
```
#(2.2)
function printNumberTypeTernary(num)
    println(iseven(num) ? "Чётное число: $num" : "Нечётное число")
end

# Пример использования тернарного оператора
printNumberTypeTernary(4)
printNumberTypeTernary(7)
printNumberType(76)
```

```
Чётное число: 4
Нечётное число
Чётное число: 76
```

3. Напишите функцию add_one, которая добавляет 1 к своему входу.

In [7]:
```
function add_one(x)
    return x + 1
end

# Пример использования
result = add_one(5)
println("Результат: $result")
```

```
Результат: 6
```

4. Используйте $map()$ или $broadcast()$ для задания матрицы $$A$$, каждый элемент которой увеличивается на единицу по сравнению с предыдущим.

In [8]:
```
# Создаем исходную матрицу A
A = [1 2 3 ; 4 5 6 ; 7 8 9]

# Используем broadcast для увеличения каждого элемента на единицу
B = broadcast(x -> x + 1, A)

# Выводим исходную и измененную матрицы
println("Исходная матрица A:")
display(A)

println("\nМатрица B (увеличенная на единицу):")
display(B)
```

```
Исходная матрица A:
3×3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9
Матрица B (увеличенная на единицу):
3×3 Matrix{Int64}:
 2  3   4
 5  6   7
 8  9  10
```

5. Задайте матрицу $$A$$ следующего вида:

A =$\begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix}\ $

– Найдите $$A$^{3}.
– Замените третий столбец матрицы $$A$$ на сумму второго и третьего столбцов.

In [9]:
```
# Задаем матрицу A
A = [1 1 3; 5 2 6; -2 -1 -3]

println("Матрица A:")
display(A)

A_cube = broadcast(x -> x^3, A)

println("\nМатрица A^3:")
display(A_cube)
println(" ")
```

```
A[:, 3] .= A[:, 2] + A[:, 3]

println("Матрица A:")
display(A)
```

```
Матрица A:
3×3 Matrix{Int64}:
  1   1   3
  5   2   6
 -2  -1  -3
Матрица A^3:
3×3 Matrix{Int64}:
   1   1   27
 125   8  216
  -8  -1  -27

Матрица A:
3×3 Matrix{Int64}:
  1   1   4
  5   2   8
 -2  -1  -4
```

6. Создайте матрицу B с элементами $B_{i1}=10,B_{i2}=-10,B_{i3}=10,i = 1,2,\cdots ,15.$ Вычислите матрицу C = $B^{T}B$.

In [10]:
```
# Создаем матрицу B
B = zeros(Int, 15, 3)
for i in 1:15
    B[i,1] = 10; B[i,2] = -10; B[i,3] = 10
end

C = transpose(B) * B

println("Матрица B:")
display(B)

println("\nМатрица C:")
display(C)
```

```
Матрица B:
15×3 Matrix{Int64}:
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
Матрица C:
```

```
3×3 Matrix{Int64}:
  1500  -1500   1500
 -1500   1500  -1500
  1500  -1500   1500
```

7. Создайте матрицу Z размерности $6 × 6$, все элементы которой равны нулю, и матрицу E, все элементы которой равны 1. Используя цикл $while$ или for и закономерности расположения элементов, создайте следующие матрицы размерности $6 × 6$:

$$Z_{1}=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}\ ,(7.1) \qquad Z_{2}=\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}\ ,(7.2) \qquad Z_{3}=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}\ ,(7.3) \qquad Z_{4}=\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}\, (7.4) $$.

In [11]:
```julia
Z = zeros(Int, 6, 6)
E = ones(Int, 6, 6)

function create_Z1(n)
    Z1 = copy(Z)
    for i in 1:n-1
        Z1[i, i+1] = 1
        Z1[i+1, i] = 1
    end
    return Z1
end

# Создаем матрицы Z1, Z2, Z3, Z4
Z1 = create_Z1(6)


# Вывод результатов
println("Матрица Z1:")
display(Z1)
```

```
Матрица Z1:
6×6 Matrix{Int64}:
 0  1  0  0  0  0
 1  0  1  0  0  0
 0  1  0  1  0  0
 0  0  1  0  1  0
 0  0  0  1  0  1
 0  0  0  0  1  0
```

In [12]:
```julia
Z = zeros(Int, 6, 6)
E = ones(Int, 6, 6)

Z2 = copy(Z)
# Создание матрицы Z2, используя закономерности расположения элементов
```

```
for i in 1:6
    for j in 1:6
        if abs(i == j) || (i == j-2) || (i == j+2)
            Z2[i, j] = 1
        end
    end
end

println("\nМатрица Z2:")
display(Z2)
```

```
Матрица Z2:
6×6 Matrix{Int64}:
 1  0  1  0  0  0
 0  1  0  1  0  0
 1  0  1  0  1  0
 0  1  0  1  0  1
 0  0  1  0  1  0
 0  0  0  1  0  1
```

In [13]:
```
Z = zeros(Int, 6, 6)
E = ones(Int, 6, 6)

Z3 = copy(Z)
# Создание матрицы Z2, используя закономерности расположения элементов
for i in 1:6
    for j in 1:6
        if abs(i+j == 7) || (i+j == 5) || (i+j == 9)
            Z3[i, j] = 1
        end
    end
end

println("\nМатрица Z3:")
display(Z3)
```

```
Матрица Z3:
6×6 Matrix{Int64}:
 0  0  0  1  0  1
 0  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  0
 1  0  1  0  0  0
```

In [14]:
```
Z = zeros(Int, 6, 6)
E = ones(Int, 6, 6)

Z4 = copy(Z)
# Создание матрицы Z2, используя закономерности расположения элементов
for i in 1:6
    for j in 1:6
        if abs((i+j) % 2 == 0)
            Z4[i, j] = 1
        end
    end
end

println("\nМатрица Z4:")
display(Z4)
```

```
Матрица Z4:
6×6 Matrix{Int64}:
 1  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  1
```

8. 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 $$A$$ и $$B$$ размерностями $$L \times M$$ и $$M \times N$$ соответственно, где элементы результирующей матрицы $$C$$ имеют вид C_{ij} =$\sum_{k=1}^{M}A_{ik}B_{jk}$ (или в тензорном виде C_{k}^{j} =$\sum_{k=1}^{M}\mathrm{A}_{k}^{i}\mathrm{B}_{j}^{k}$).
   – Используя написанную вами функцию $outer()$, создайте матрицы следующей структуры:
   $$A_{1}=\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{pmatrix} ,\qquad A_{2}=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 16 & 32 \\ 3 & 9 & 27 & 81 & 243 \\ 4 & 16 & 64 & 256 & 1024 \end{pmatrix} ,\qquad A_{3}=\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{pmatrix} $$,
   $$A_{4}=\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} , \qquad A_{5}=\begin{pmatrix} 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 \\ 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 \\ 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \end{pmatrix} $$.

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

In [15]:
```
# Функция outer аналогичная функции outer в R
function outer(x, y, operation)
    L, M = size(x)
    M, N = size(y)

    result = zeros(eltype(x), L, N)

    for i in 1:L
        for j in 1:N
            result[i, j] = sum(operation(x[i, k], y[k, j]) for k in 1:M)
```

```
        end
    end

    return result
end

# Создание матриц A1, A2, A3, A4, A5 с использованием outer
A1 = outer(reshape(0:4, 5, 1), reshape(0:4, 1, 5), +)
A2 = outer(reshape(0:4, 5, 1), reshape(1:5, 1, 5), ^)
A3 = outer(hcat([[if i==j 1 else 0 end for j in 0:4] for i in 0:4]...), hcat([Ve
A4 = outer(hcat([[if i==j 1 else 0 end for j in 0:9] for i in 0:9]...), hcat([Ve
A5 = outer(hcat([[if i==j 1 else 0 end for j in 0:8] for i in 0:8]...), hcat([Ve

# Вывод результатов
println("A1:")
display(A1)

println("\nA2:")
display(A2)

println("\nA3:")
display(A3)

println("\nA4:")
display(A4)

println("\nA5:")
display(A5)
```

```
A1:
5×5 Matrix{Int64}:
 0  1  2  3  4
 1  2  3  4  5
 2  3  4  5  6
 3  4  5  6  7
 4  5  6  7  8
A2:
5×5 Matrix{Int64}:
 0   0   0    0     0
 1   1   1    1     1
 2   4   8   16    32
 3   9  27   81   243
 4  16  64  256  1024
A3:
5×5 Matrix{Int64}:
 0  1  2  3  4
 1  2  3  4  0
 2  3  4  0  1
 3  4  0  1  2
 4  0  1  2  3
A4:
```

```
10×10 Matrix{Int64}:
 0  1  2  3  4  5  6  7  8  9
 1  2  3  4  5  6  7  8  9  0
 2  3  4  5  6  7  8  9  0  1
 3  4  5  6  7  8  9  0  1  2
 4  5  6  7  8  9  0  1  2  3
 5  6  7  8  9  0  1  2  3  4
 6  7  8  9  0  1  2  3  4  5
 7  8  9  0  1  2  3  4  5  6
 8  9  0  1  2  3  4  5  6  7
 9  0  1  2  3  4  5  6  7  8
A5:
9×9 Matrix{Int64}:
 0  1  2  3  4  5  6  7  8
 8  0  1  2  3  4  5  6  7
 7  8  0  1  2  3  4  5  6
 6  7  8  0  1  2  3  4  5
 5  6  7  8  0  1  2  3  4
 4  5  6  7  8  0  1  2  3
 3  4  5  6  7  8  0  1  2
 2  3  4  5  6  7  8  0  1
 1  2  3  4  5  6  7  8  0
```

9. Решите следующую систему линейных уравнений с 5 неизвестными:

$\begin{cases} x_{1}+2x_{2}+3x_{3}+4x_{4}+5x_{5}=7\\ 2x_{1}+x_{2}+2x_{3}+3x_{4}+4x_{5}=-1\\ 3x_{1}+2x_{2}+x_{3}+2x_{4}+3x_{5}=-3\\ 4x_{1}+3x_{2}+2x_{3}+x_{4}+2x_{5}=5\\ 5x_{1}+4x_{2}+3x_{3}+2x_{4}+x_{5}=17\\ \end{cases} $

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

In [16]:
```
M = [1. 2. 3. 4. 5.; 2. 1. 2. 3. 4.; 3. 2. 1. 2. 3.; 4. 3. 2. 1. 2.; 5. 4. 3. 2.
v = [7. ,-1. ,-3. ,5. ,17.] # Вектор (правая часть системы)

x = M \ v

println("Матрица коэффициентов:")
display(M)
println("Вектор решений:")
display(v)
println("Вектор ответ:")
println(x)
```

```
Матрица коэффициентов:
5×5 Matrix{Float64}:
 1.0  2.0  3.0  4.0  5.0
 2.0  1.0  2.0  3.0  4.0
 3.0  2.0  1.0  2.0  3.0
 4.0  3.0  2.0  1.0  2.0
 5.0  4.0  3.0  2.0  1.0
Вектор решений:
```

```
5-element Vector{Float64}:
  7.0
 -1.0
 -3.0
  5.0
 17.0
Вектор ответ:
[-2.000000000000036, 3.0000000000000058, 4.999999999999998, 1.9999999999999991,
-3.999999999999999]
```

10. Создайте матрицу M размерности $6 × 10$, элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности $1, 2, … , 10$.
    – Найдите число элементов в каждой строке матрицы M, которые больше числа N (например, $N = 4$).
    – Определите, в каких строках матрицы M число M(например,$M = 7$) встречается ровно 2 раза?
    – Определите все пары столбцов матрицы M, сумма элементов которых больше K (например, $K = 75$).

In [17]:
```
using Random

Random.seed!(30)

rows = 6
cols = 10

M = rand(1:10, rows, cols)
N = 4

count_elements_greater_than_N = sum(M .> N, dims=2)

println("Матрица M:")
display(M)
println("Число элементов в каждой строке, больших $N:")
println(count_elements_greater_than_N)
```

```
Матрица M:
6×10 Matrix{Int64}:
  8   2  8   7  10   4  2  9   6  2
  7   4  9   8   6  10  5  9   4  4
 10   8  7  10   4   1  1  6  10  9
  4   4  5  10   9   9  1  7   7  6
  9   8  8   9   6   7  3  8   4  4
  9  10  6   8   6   4  3  6   6  9
Число элементов в каждой строке, больших 4:
[6; 7; 7; 7; 7; 8;;]
```

In [18]:
```
using Random

Random.seed!(30)

rows = 6
cols = 10

M = rand(1:10, rows, cols)
```

```
M_value = 7

rows_with_2_occurrences = findall(x -> count(isequal(M_value), x) == 2, eachrow(

println("Матрица M:")
display(M)
println("Строки, в которых число M встречается ровно 2 раза:")
println(rows_with_2_occurrences)
```

```
Матрица M:
6×10 Matrix{Int64}:
  8   2  8   7  10   4  2  9   6  2
  7   4  9   8   6  10  5  9   4  4
 10   8  7  10   4   1  1  6  10  9
  4   4  5  10   9   9  1  7   7  6
  9   8  8   9   6   7  3  8   4  4
  9  10  6   8   6   4  3  6   6  9
Строки, в которых число M встречается ровно 2 раза:
[4]
```

In [19]:
```
using Random

Random.seed!(30)

rows = 6
cols = 10

M = rand(1:10, rows, cols)

K = 75

column_pairs = [(i, j) for i in 1:cols-1 for j in i+1:cols if sum(M[:, i] + M[:,

println("Матрица M:")
display(M)
println("Пары столбцов, сумма элементов которых больше $K:")
println(column_pairs)
```

```
Матрица M:
6×10 Matrix{Int64}:
  8   2  8   7  10   4  2  9   6  2
  7   4  9   8   6  10  5  9   4  4
 10   8  7  10   4   1  1  6  10  9
  4   4  5  10   9   9  1  7   7  6
  9   8  8   9   6   7  3  8   4  4
  9  10  6   8   6   4  3  6   6  9
Пары столбцов, сумма элементов которых больше 75:
[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 8), (1, 9), (1, 10), (2, 3), (2, 4),
(2, 5), (2, 8), (3, 4), (3, 5), (3, 6), (3, 8), (3, 9), (3, 10), (4, 5), (4, 6),
(4, 8), (4, 9), (4, 10), (5, 6), (5, 8), (5, 9), (6, 8), (8, 9), (8, 10)]
```

11. Вычислите:

    -$\sum_{i=1}^{20}\sum_{j=1}^{5}\frac{i^4}{(3+j)}$,

    -$\sum_{i=1}^{20}\sum_{j=1}^{5}\frac{i^4}{(3+ij)}$.

In [20]:
```
result = 0.0

for i in 1:20
    for j in 1:5
```

```
        result += i^4 / (3 + j)
    end
end

println("Результат вычисления суммы: ", result)
```

```
Результат вычисления суммы: 639215.2833333334
```

In [21]:
```
result = 0.0

for i in 1:20
    for j in 1:5
        result += i^4 / (3 + i*j)
    end
end

println("Результат вычисления суммы: ", result)
```

```
Результат вычисления суммы: 89912.02146097136
```

Выводы:

Мною освоены применения циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

Список литературы:

Julia 1.5 Documentation. — 2020. — URL: https://docs.julialang.org/en/v1/.\ Klok H.,Nazarathy Y. Statistics with Julia: Fundamentals for Data Science,Machine Learning and Artificial Intelligence. — 2020. — URL: https://statisticswithjulia.org/.\ Ökten G. First Semester in Numerical Analysis with Julia. — Florida State University, 2019. — DOI: 10.33009/jul.
Антонюк В. А. Язык Julia как инструмент исследователя. — М. : Физический факультет МГУ им. М. В. Ломоносова, 2019.
Шиндин А. В. Язык программирования математических вычислений Julia. Базовое руководство. — Нижний Новгород : Нижегородский госуниверситет, 2016.
Задание лабораторной работы №3 - https://esystem.rudn.ru/pluginfile.php/2231400/mod_resource/content/2/003-lab_control-structures.pdf