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 математическому моделированию

Студент: Яссин Мохамад Аламин

Группа: НКНбд-01-20

МОСКВА

2023 г.

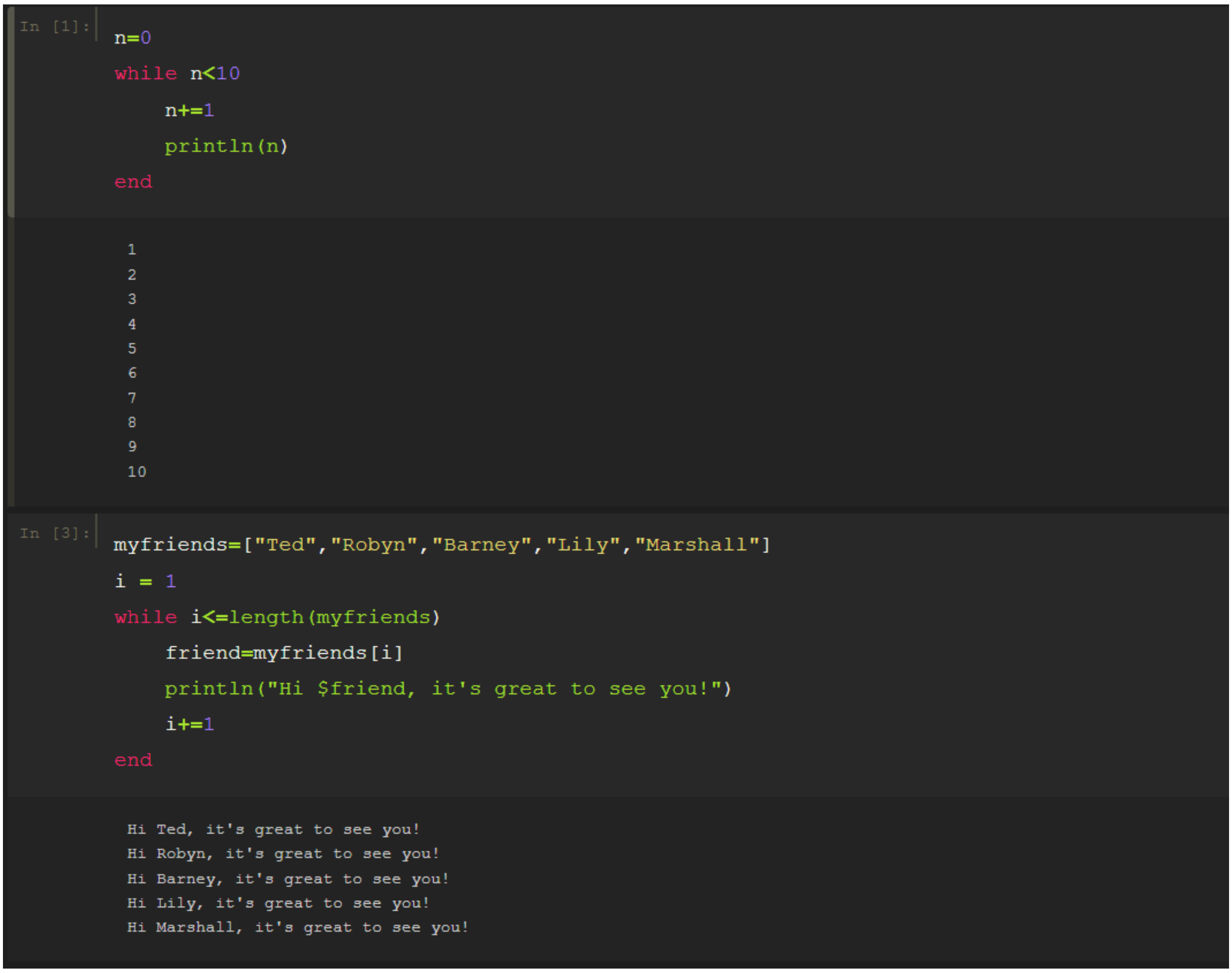
Постановка задачи

Основная цель работы — изучить несколько структур данных, реализованных в Julia, научиться применять их и операции над ними для решения задач.

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

Выполнение работы

1. Повторение примером



In [1]:

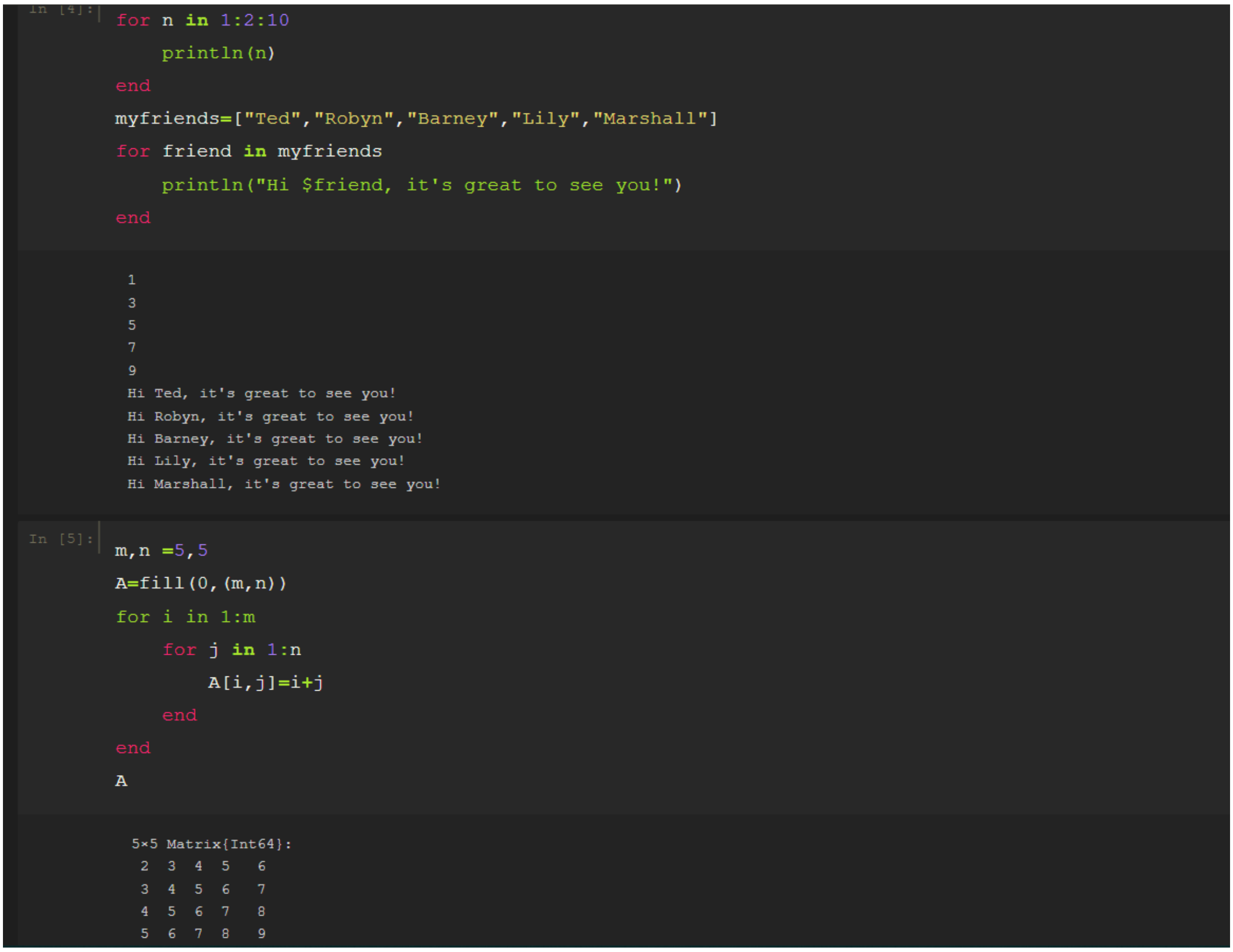
```
n=0
while n<10
    n+=1
    println(n)
end
```

```
1
2
3
4
5
6
7
8
9
10
```

In [3]:

```
myfriends=["Ted","Robyn","Barney","Lily","Marshall"]
i = 1
while i<=length(myfriends)
    friend=myfriends[i]
    println("Hi $friend, it's great to see you!")
    i+=1
end
```

```
Hi Ted, it's great to see you!
Hi Robyn, it's great to see you!
Hi Barney, it's great to see you!
Hi Lily, it's great to see you!
Hi Marshall, it's great to see you!
```

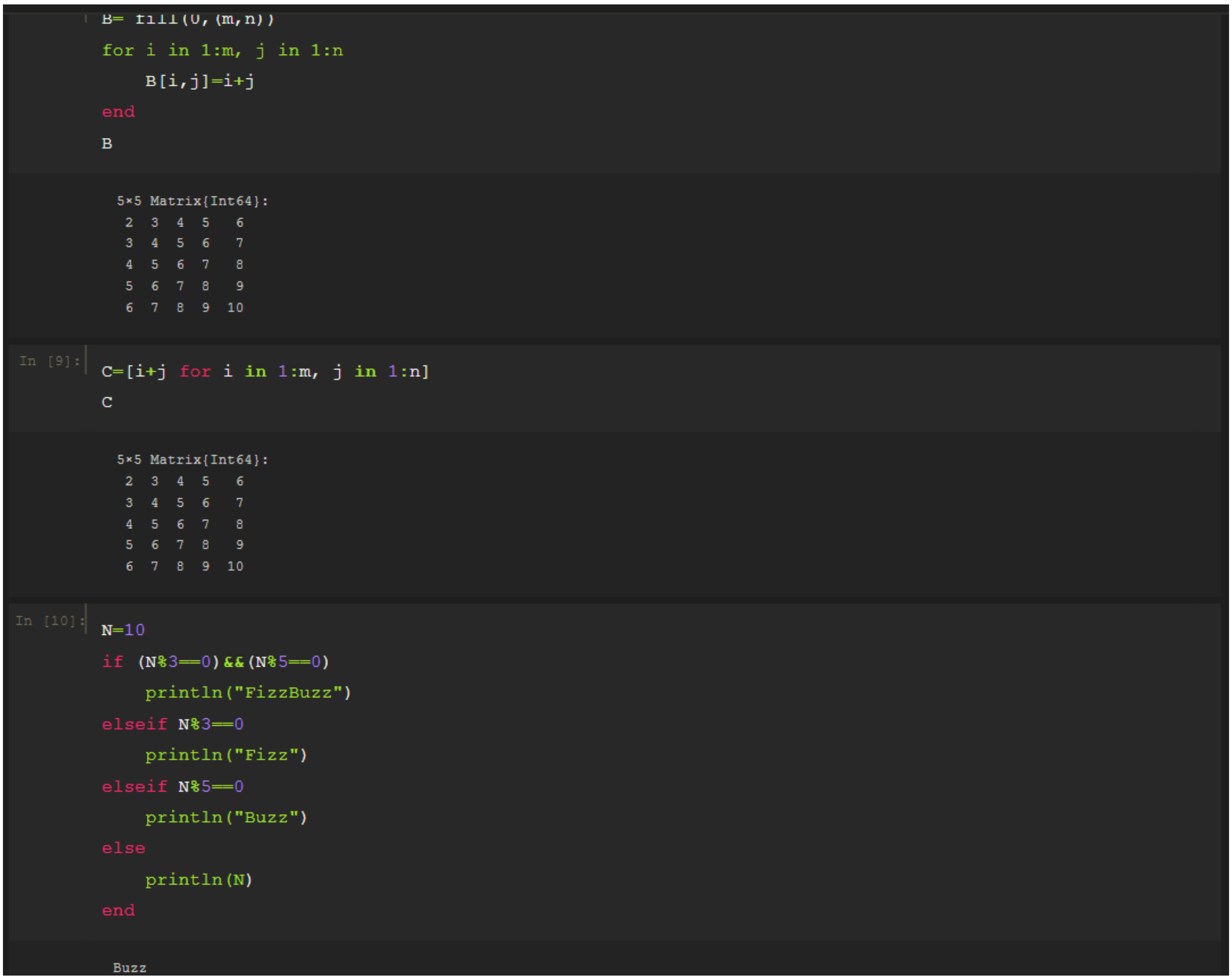


```
for n in 1:2:10
    println(n)
end
myfriends=["Ted","Robyn","Barney","Lily","Marshall"]
for friend in myfriends
    println("Hi $friend, it's great to see you!")
end
```

```
1
3
5
7
9
Hi Ted, it's great to see you!
Hi Robyn, it's great to see you!
Hi Barney, it's great to see you!
Hi Lily, it's great to see you!
Hi Marshall, it's great to see you!
```

```
m,n =5,5
A=fill(0,(m,n))
for i in 1:m
    for j in 1:n
        A[i,j]=i+j
    end
end
A
```

```
5×5 Matrix{Int64}:
 2  3  4  5   6
 3  4  5  6   7
 4  5  6  7   8
 5  6  7  8   9
```



```
B= fill(0,(m,n))
for i in 1:m, j in 1:n
    B[i,j]=i+j
end
B
```

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

In [9]:

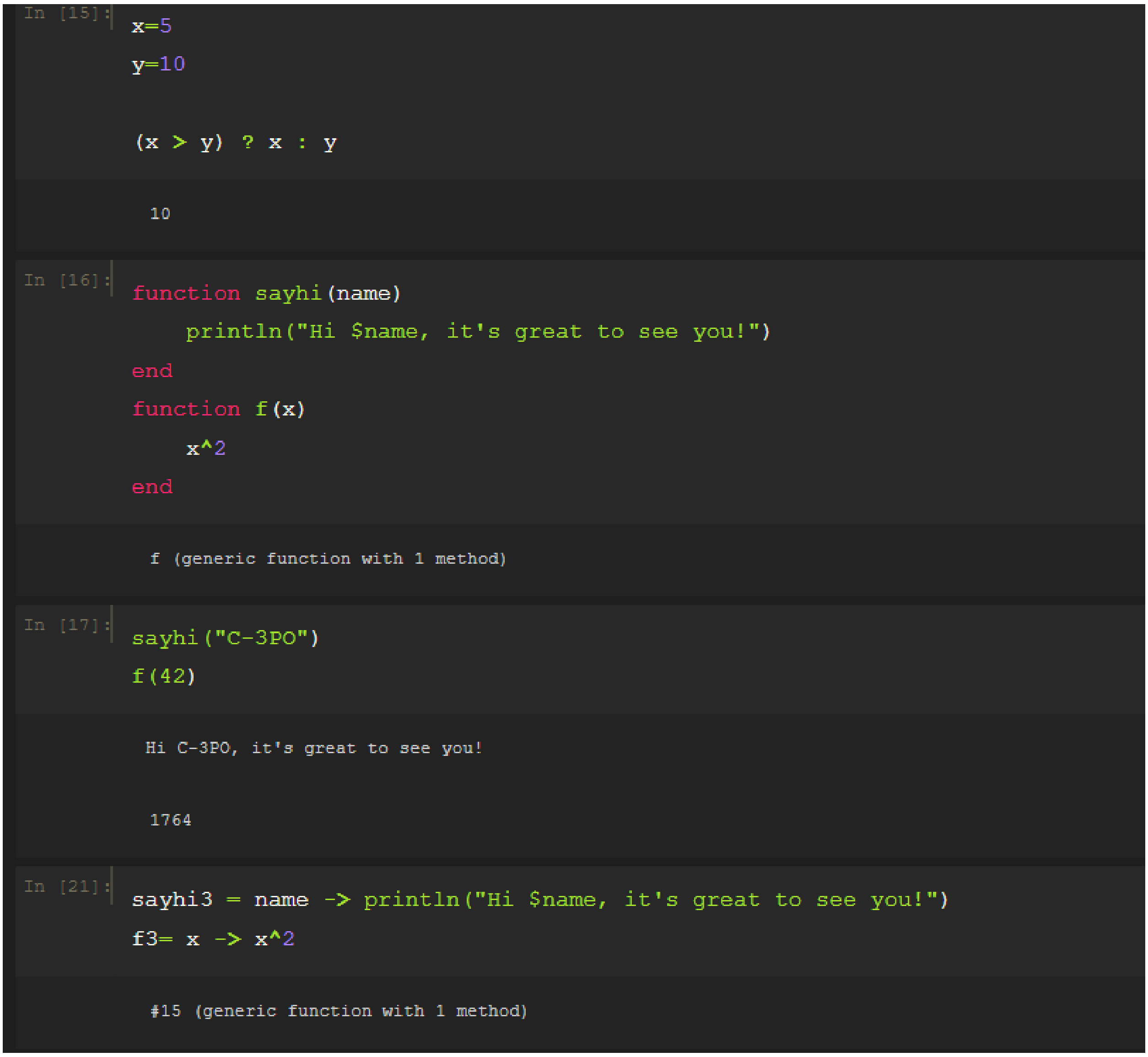
```
C=[i+j for i in 1:m, j in 1:n]
C
```

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

In [10]:

```
N=10
if (N%3==0)&&(N%5==0)
    println("FizzBuzz")
elseif N%3==0
    println("Fizz")
elseif N%5==0
    println("Buzz")
else
    println(N)
end
```

```
Buzz
```



In [15]:

```
x=5
y=10

(x > y) ? x : y
```

```
10
```

In [16]:

```
function sayhi(name)
    println("Hi $name, it's great to see you!")
end
function f(x)
    x^2
end
```

```
f (generic function with 1 method)
```

In [17]:

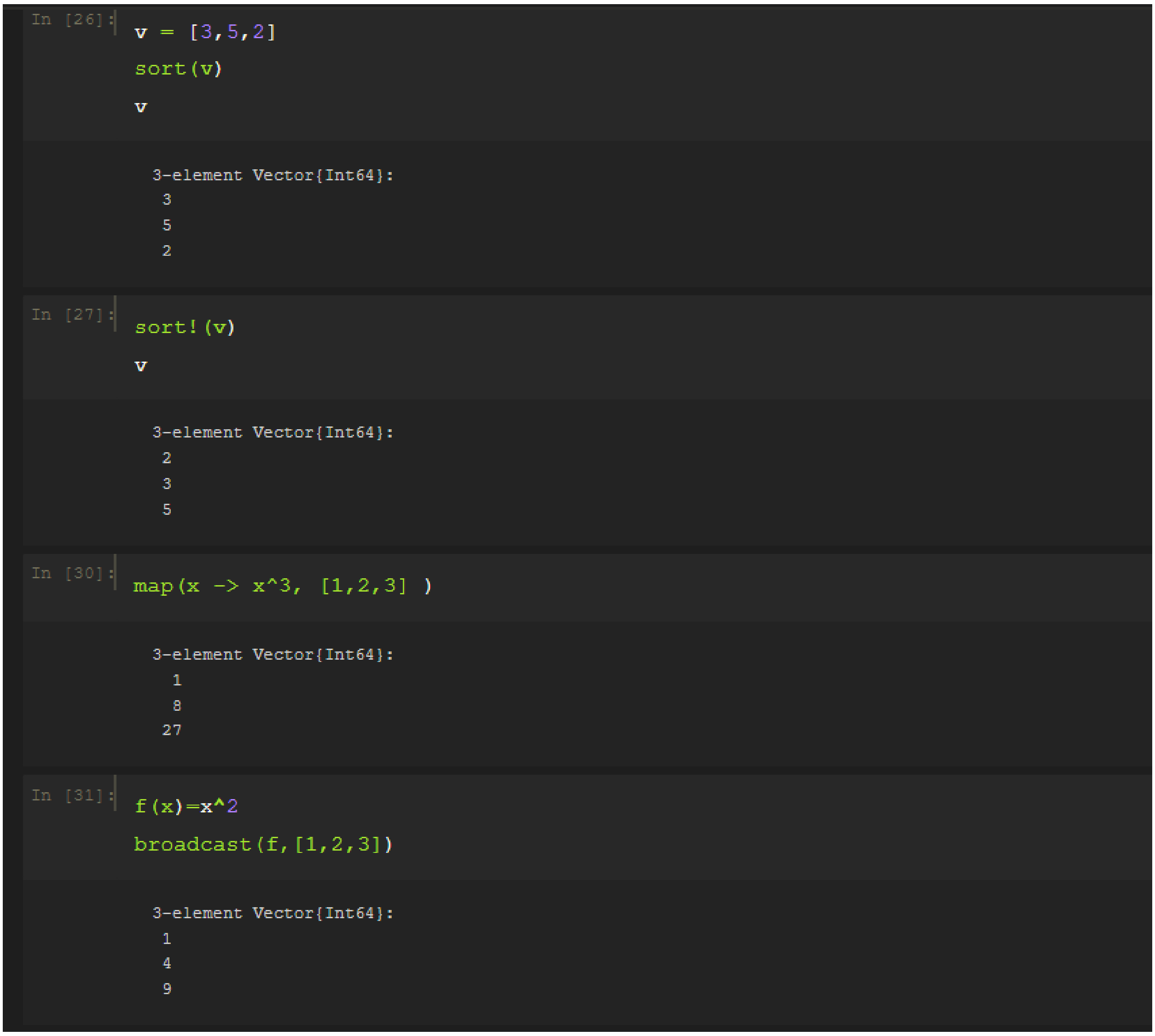
```
sayhi("C-3PO")
f(42)
```

```
Hi C-3PO, it's great to see you!

1764
```

In [21]:

```
sayhi3 = name -> println("Hi $name, it's great to see you!")
f3= x -> x^2
```

```
#15 (generic function with 1 method)
```



```
In [26]:
v = [3,5,2]
sort(v)
v
```

```
3-element Vector{Int64}:
 3
 5
 2
```

```
In [27]:
sort!(v)
v
```

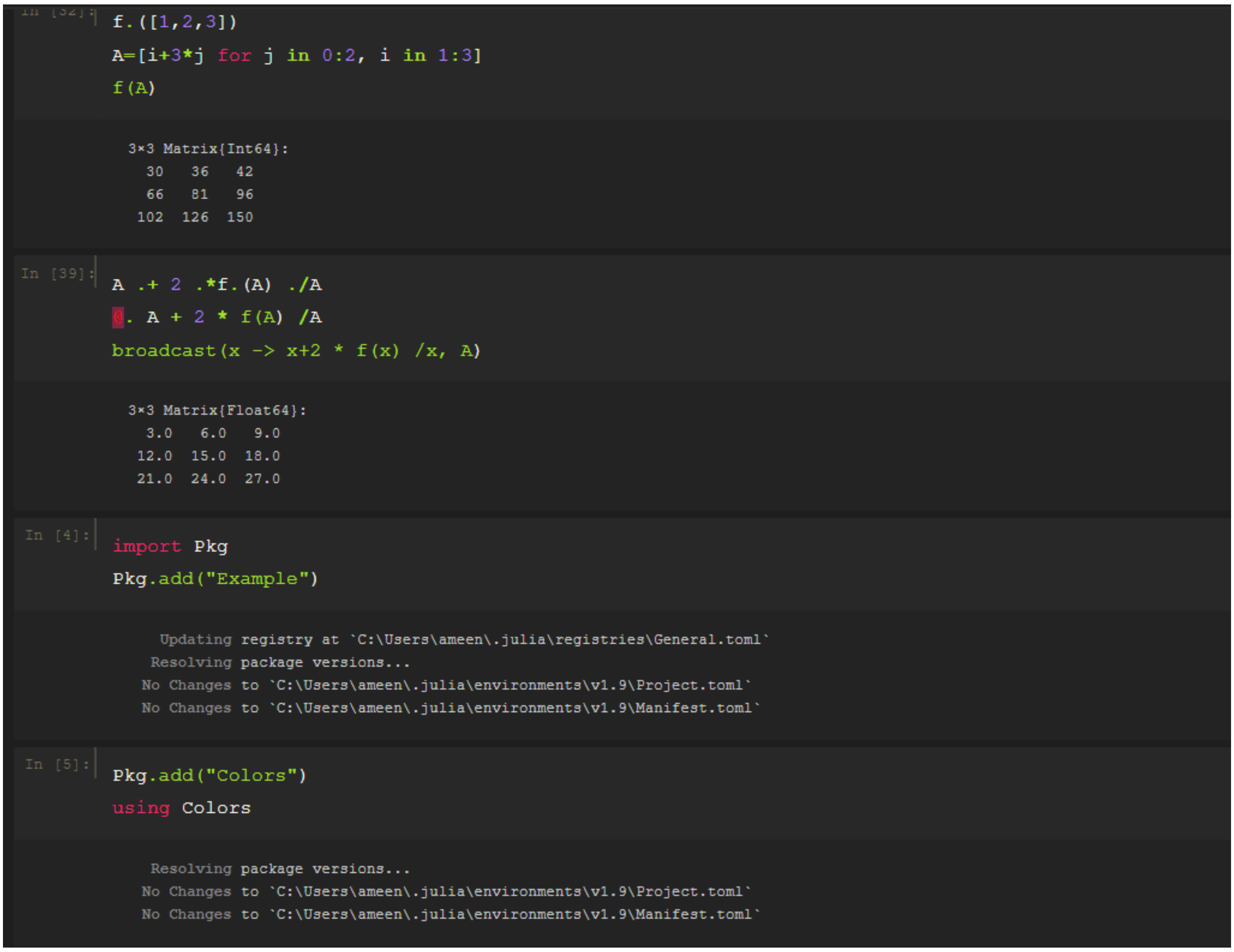
```
3-element Vector{Int64}:
 2
 3
 5
```

```
In [30]:
map(x -> x^3, [1,2,3] )
```

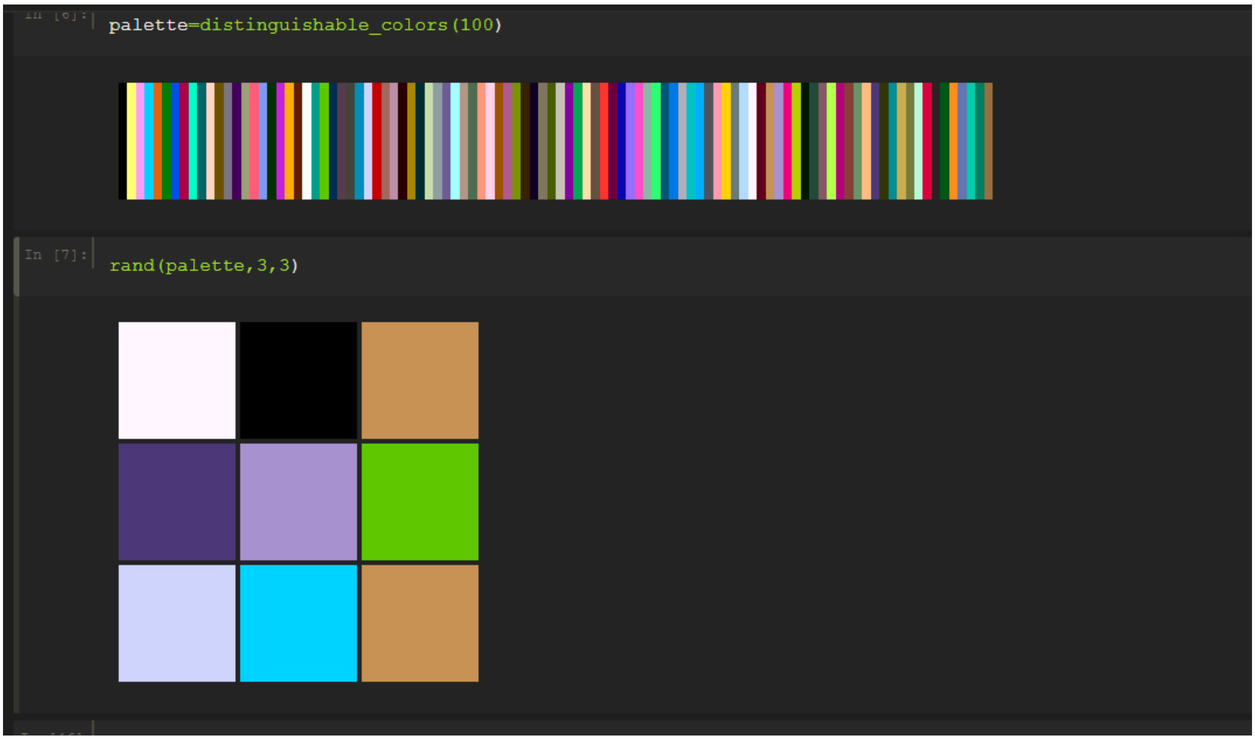
```
3-element Vector{Int64}:
  1
  8
 27
```

```
In [31]:
f(x)=x^2
broadcast(f,[1,2,3])
```

```
3-element Vector{Int64}:
 1
 4
 9
```



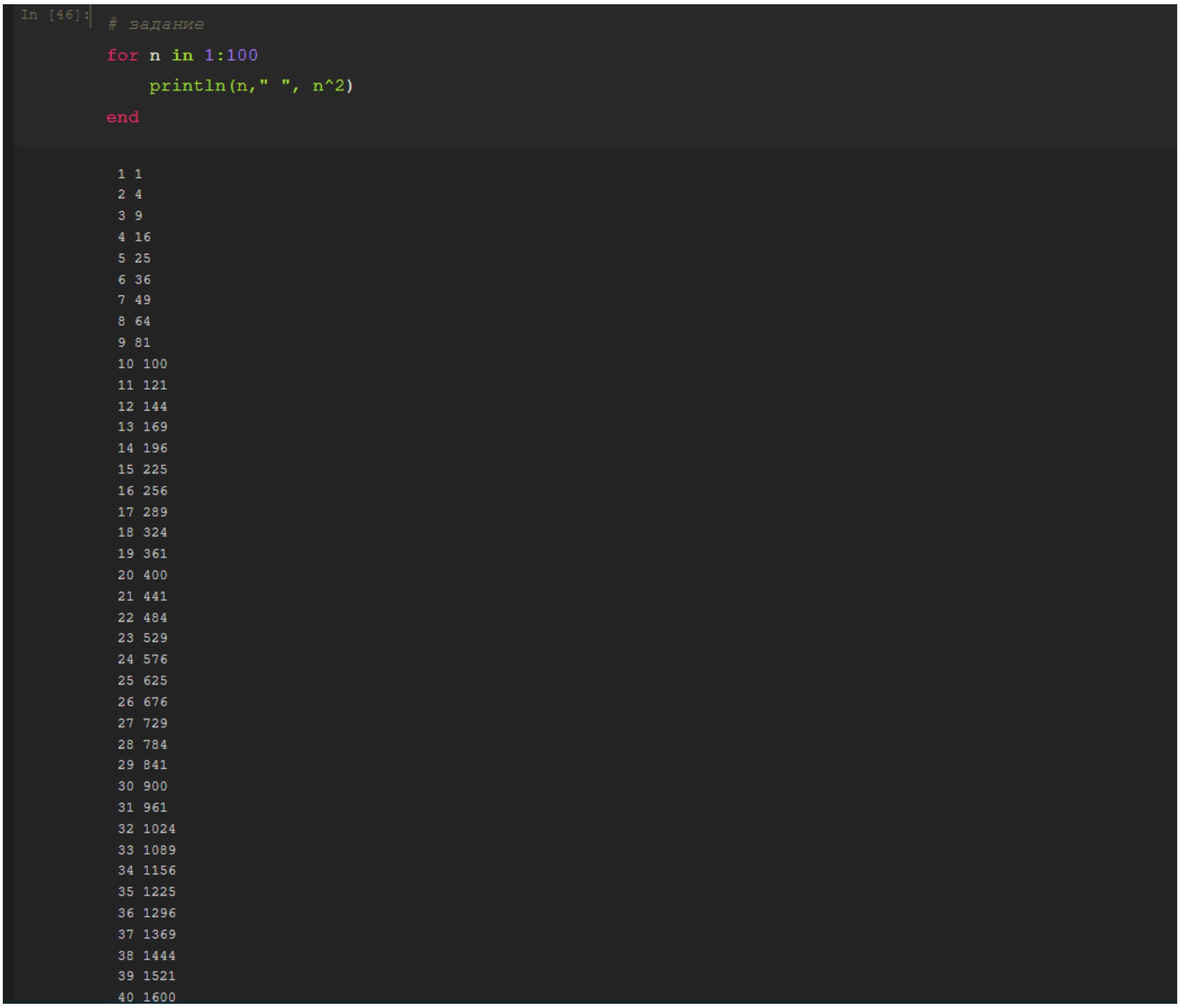
```
In [32]: f.([1,2,3])
         A=[i+3*j for j in 0:2, i in 1:3]
         f(A)

           3×3 Matrix{Int64}:
             30   36   42
             66   81   96
            102  126  150

In [39]: A .+ 2 .*f.(A) ./A
         @. A + 2 * f(A) /A
         broadcast(x -> x+2 * f(x) /x, A)

           3×3 Matrix{Float64}:
             3.0   6.0   9.0
            12.0  15.0  18.0
            21.0  24.0  27.0

In [4]:  import Pkg
         Pkg.add("Example")

              Updating registry at `C:\Users\ameen\.julia\registries\General.toml`
             Resolving package versions...
            No Changes to `C:\Users\ameen\.julia\environments\v1.9\Project.toml`
            No Changes to `C:\Users\ameen\.julia\environments\v1.9\Manifest.toml`

In [5]:  Pkg.add("Colors")
         using Colors

             Resolving package versions...
            No Changes to `C:\Users\ameen\.julia\environments\v1.9\Project.toml`
            No Changes to `C:\Users\ameen\.julia\environments\v1.9\Manifest.toml`
```



```
palette=distinguishable_colors(100)
```

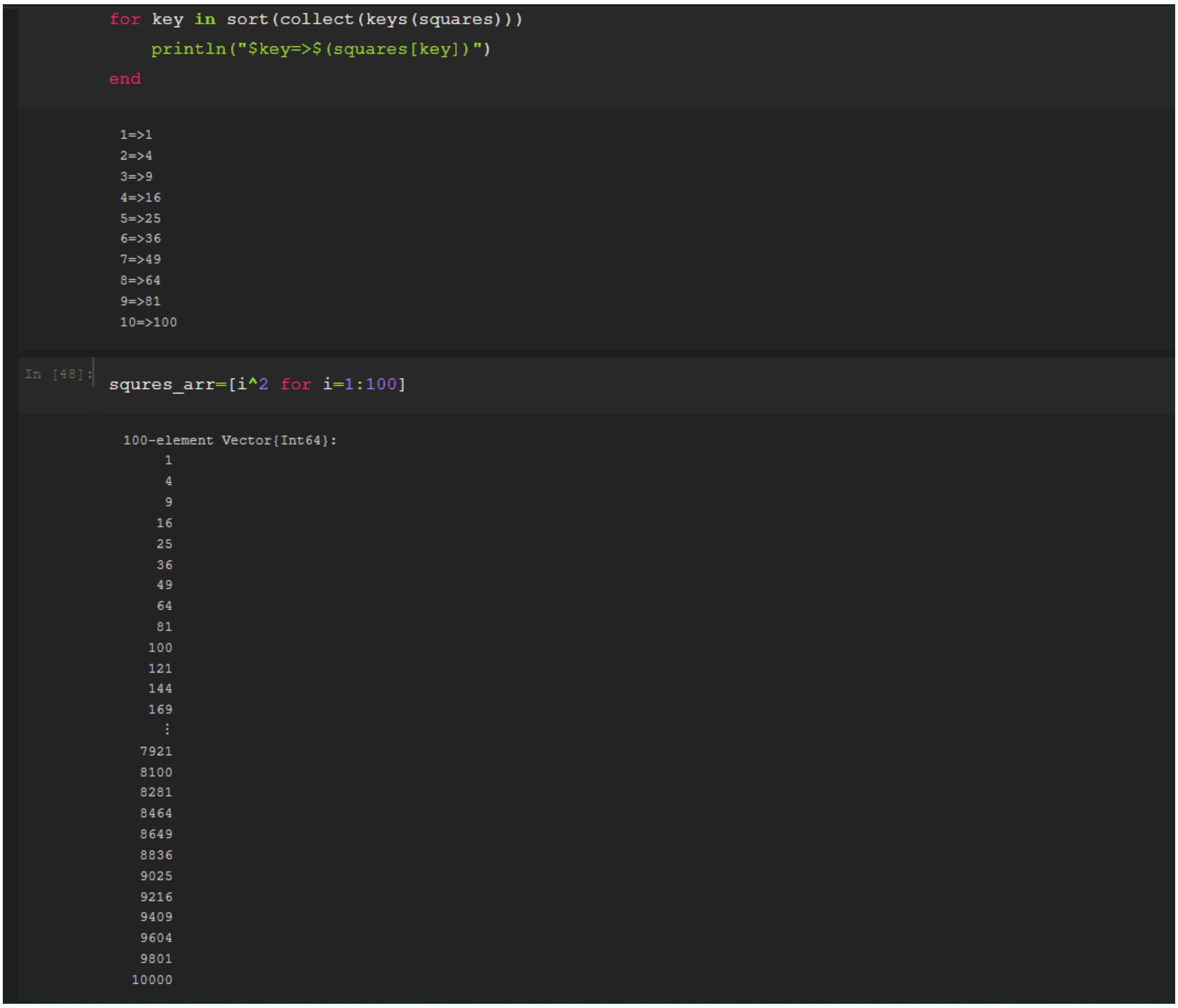
```
In [7]:  rand(palette,3,3)
```

2.Выполните задания для самостоятельной работы



```
In [46]:  # задание
           for n in 1:100
               println(n," ", n^2)
           end
```

```
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
11 121
12 144
13 169
14 196
15 225
16 256
17 289
18 324
19 361
20 400
21 441
22 484
23 529
24 576
25 625
26 676
27 729
28 784
29 841
30 900
31 961
32 1024
33 1089
34 1156
35 1225
36 1296
37 1369
38 1444
39 1521
40 1600
```

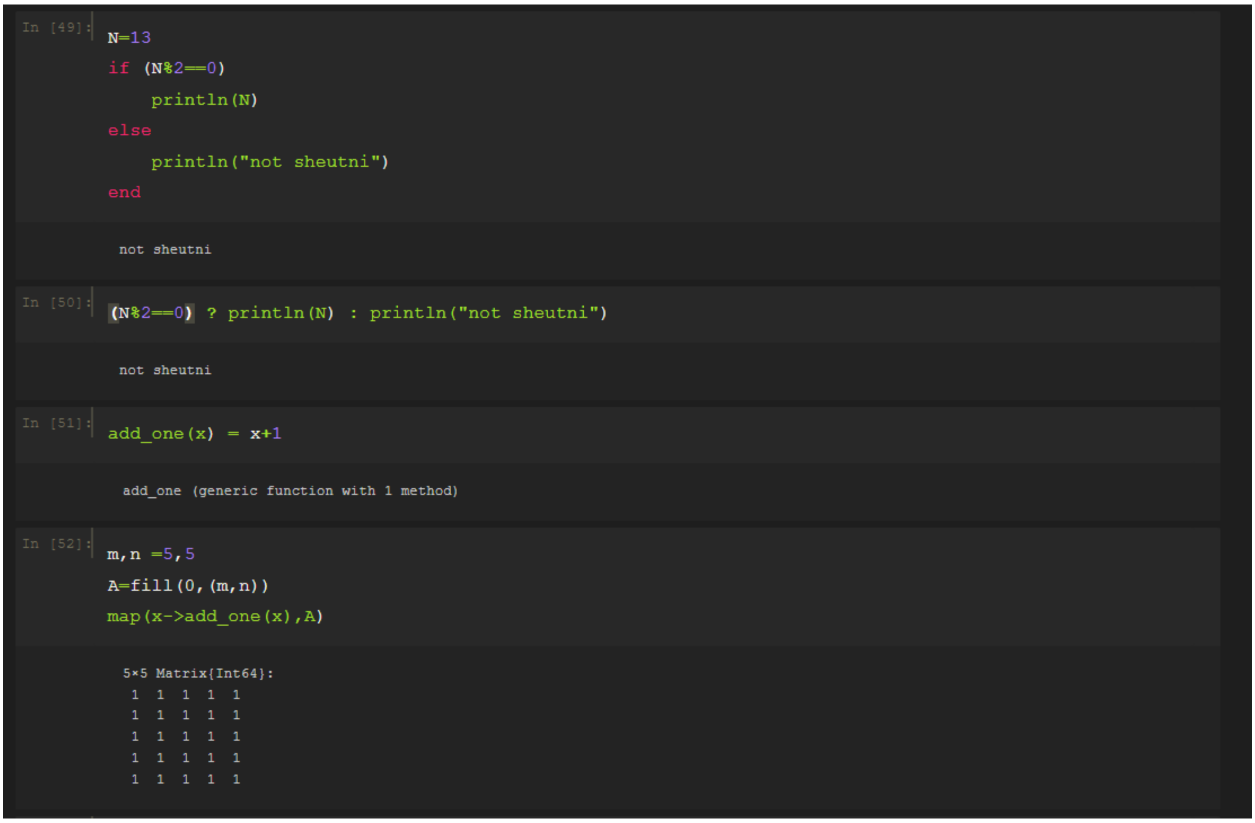


```
for key in sort(collect(keys(squares)))
    println("$key=>$(squares[key])")
end
```

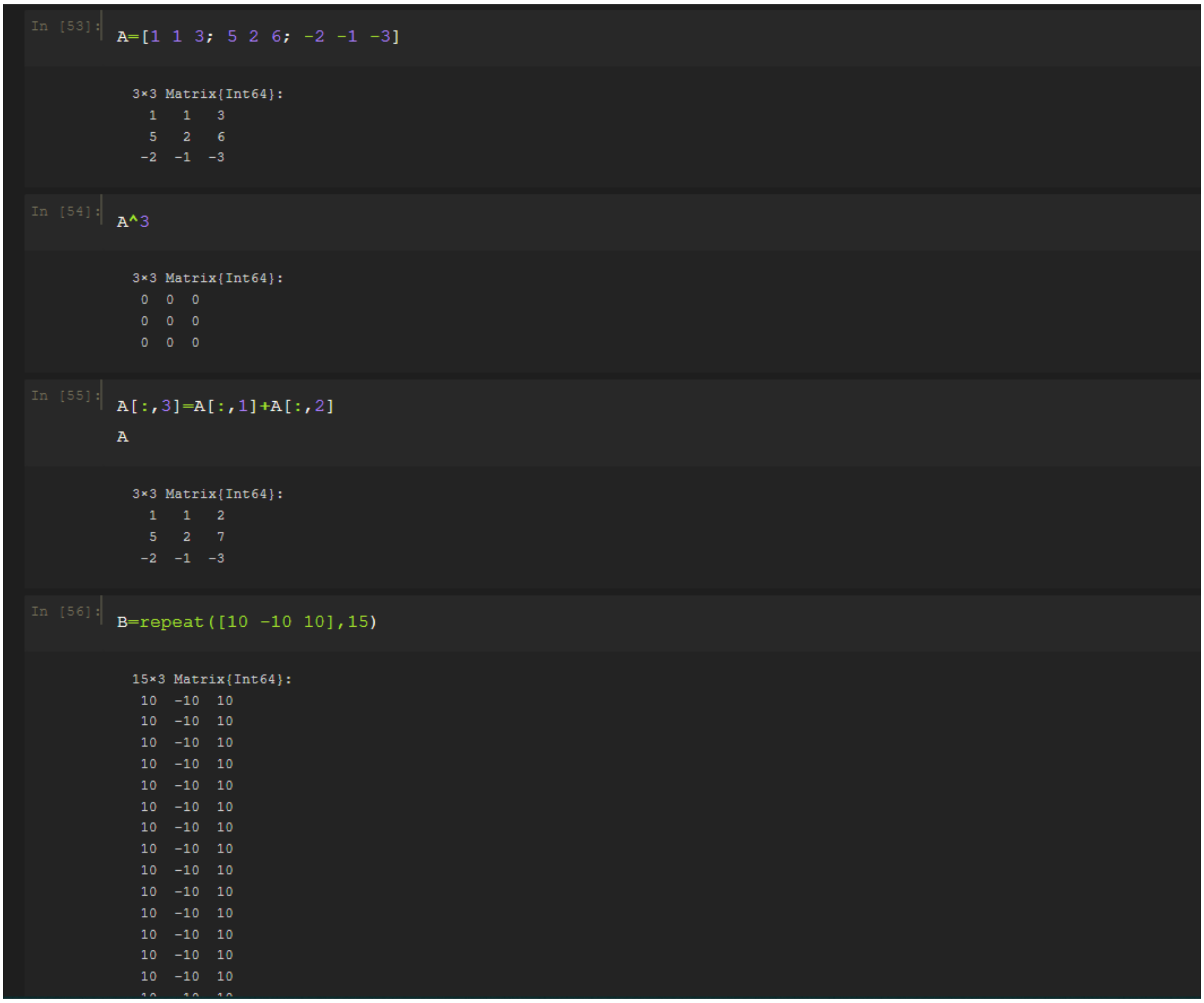
```
1=>1
2=>4
3=>9
4=>16
5=>25
6=>36
7=>49
8=>64
9=>81
10=>100
```

```
In [48]: squres_arr=[i^2 for i=1:100]
```

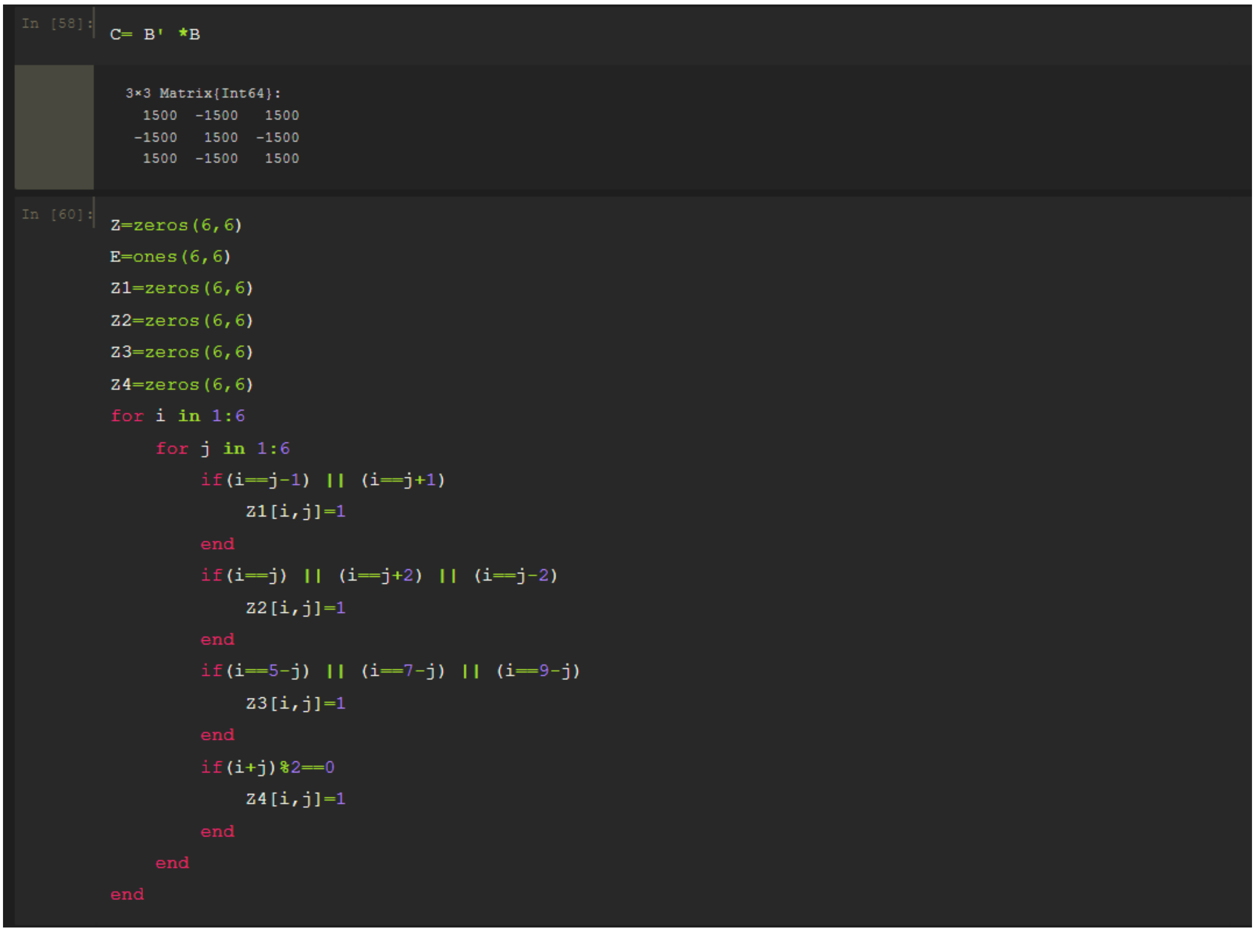
```
100-element Vector{Int64}:
     1
     4
     9
    16
    25
    36
    49
    64
    81
   100
   121
   144
   169
     ⋮
  7921
  8100
  8281
  8464
  8649
  8836
  9025
  9216
  9409
  9604
  9801
 10000
```



```
In [49]: N=13
         if (N%2==0)
             println(N)
         else
             println("not sheutni")
         end

          not sheutni

In [50]: (N%2==0) ? println(N) : println("not sheutni")

          not sheutni

In [51]: add_one(x) = x+1

          add_one (generic function with 1 method)

In [52]: m,n =5,5
         A=fill(0,(m,n))
         map(x->add_one(x),A)

          5×5 Matrix{Int64}:
           1  1  1  1  1
           1  1  1  1  1
           1  1  1  1  1
           1  1  1  1  1
           1  1  1  1  1
```



```
In [53]: A=[1 1 3; 5 2 6; -2 -1 -3]

          3×3 Matrix{Int64}:
            1   1   3
            5   2   6
           -2  -1  -3

In [54]: A^3

          3×3 Matrix{Int64}:
           0  0  0
           0  0  0
           0  0  0

In [55]: A[:,3]=A[:,1]+A[:,2]
         A

          3×3 Matrix{Int64}:
            1   1   2
            5   2   7
           -2  -1  -3

In [56]: B=repeat([10 -10 10],15)

          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
```



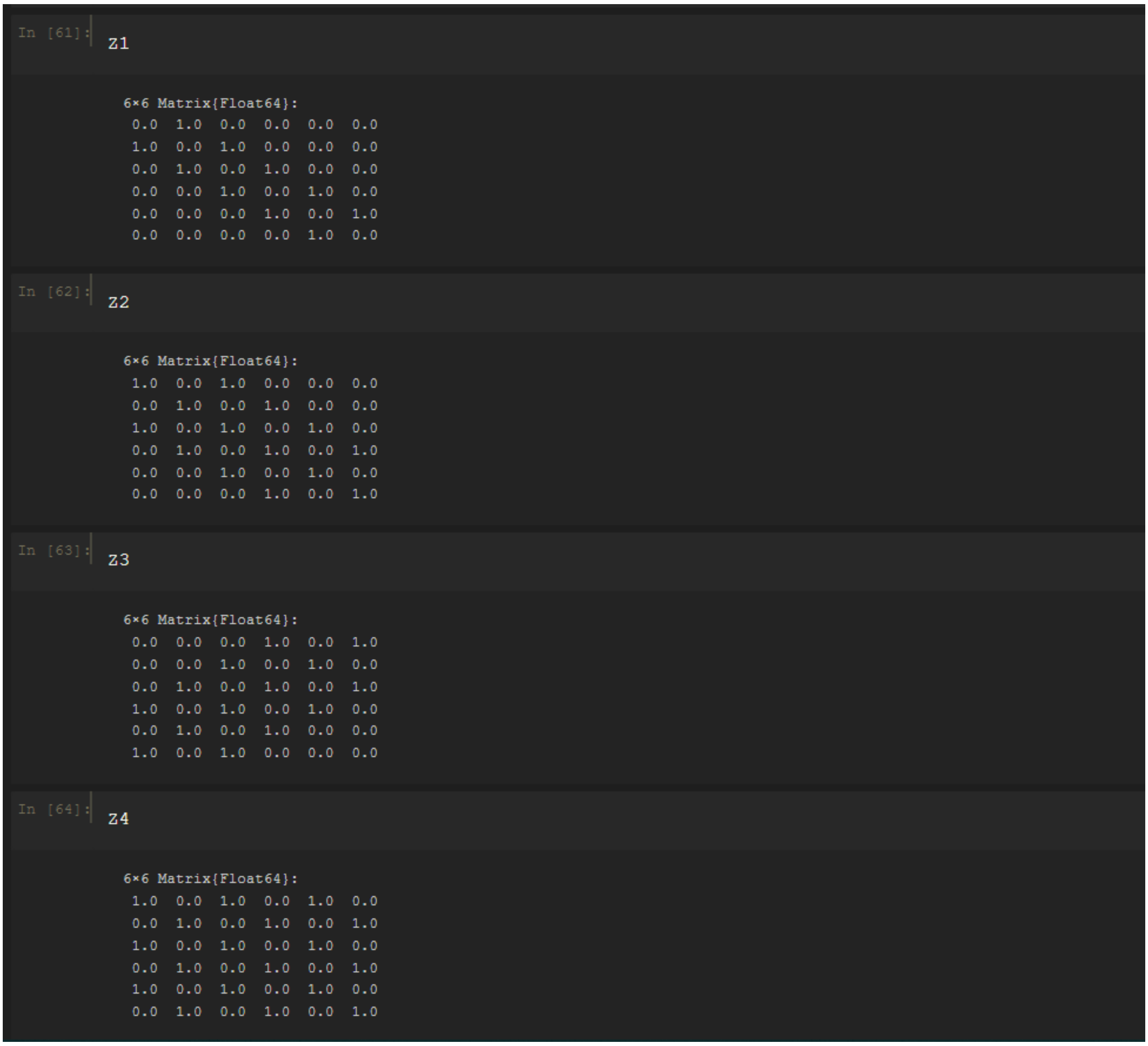
In [58]:

```
C= B' *B
```

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

In [60]:

```
Z=zeros(6,6)
E=ones(6,6)
Z1=zeros(6,6)
Z2=zeros(6,6)
Z3=zeros(6,6)
Z4=zeros(6,6)
for i in 1:6
    for j in 1:6
        if(i==j-1) || (i==j+1)
            Z1[i,j]=1
        end
        if(i==j) || (i==j+2) || (i==j-2)
            Z2[i,j]=1
        end
        if(i==5-j) || (i==7-j) || (i==9-j)
            Z3[i,j]=1
        end
        if(i+j)%2==0
            Z4[i,j]=1
        end
    end
end
```

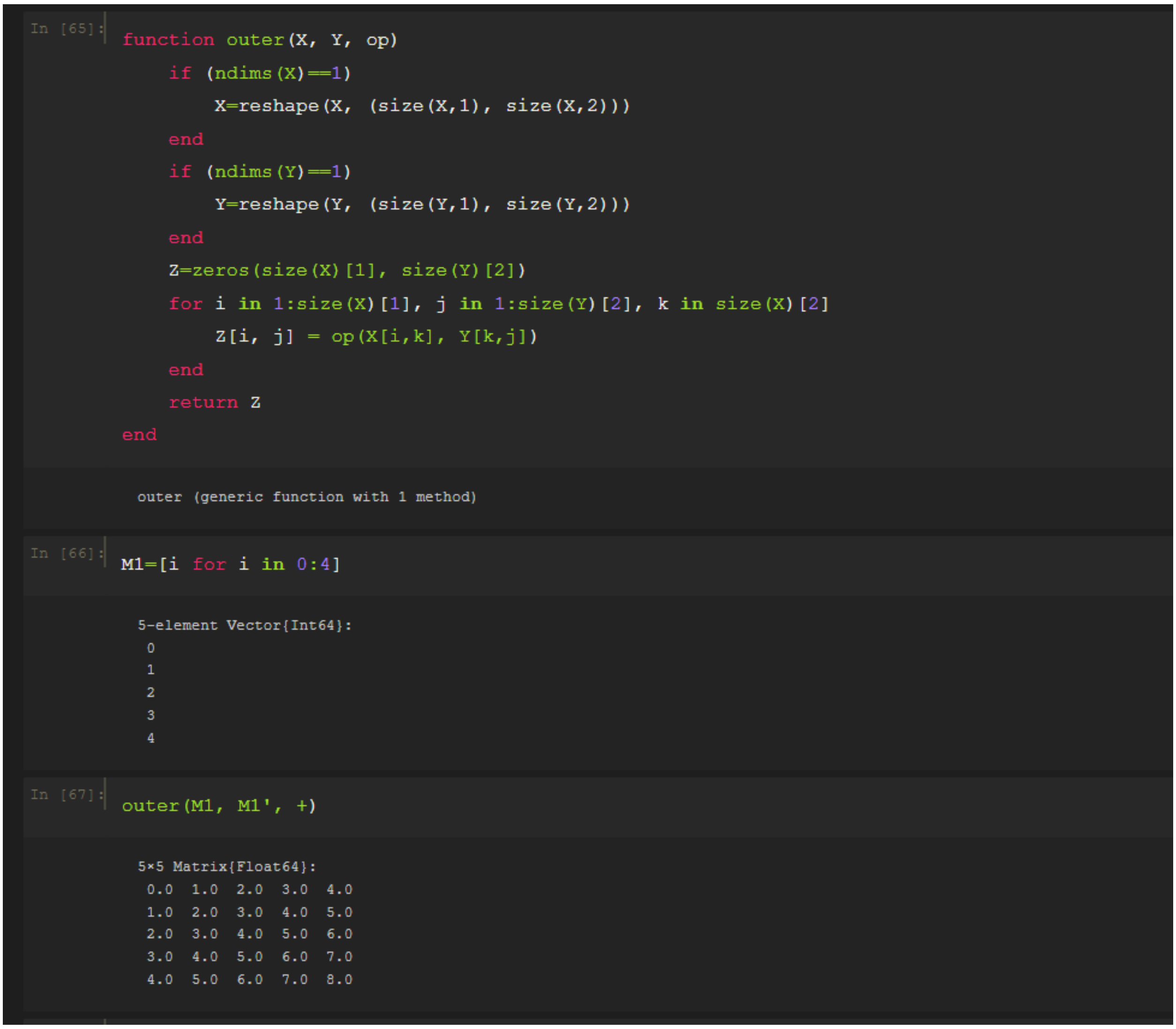


```
In [61]: Z1

6×6 Matrix{Float64}:
 0.0  1.0  0.0  0.0  0.0  0.0
 1.0  0.0  1.0  0.0  0.0  0.0
 0.0  1.0  0.0  1.0  0.0  0.0
 0.0  0.0  1.0  0.0  1.0  0.0
 0.0  0.0  0.0  1.0  0.0  1.0
 0.0  0.0  0.0  0.0  1.0  0.0
```

```
In [62]: Z2

6×6 Matrix{Float64}:
 1.0  0.0  1.0  0.0  0.0  0.0
 0.0  1.0  0.0  1.0  0.0  0.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 0.0  0.0  1.0  0.0  1.0  0.0
 0.0  0.0  0.0  1.0  0.0  1.0
```

```
In [63]: Z3

6×6 Matrix{Float64}:
 0.0  0.0  0.0  1.0  0.0  1.0
 0.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  0.0
 1.0  0.0  1.0  0.0  0.0  0.0
```

```
In [64]: Z4

6×6 Matrix{Float64}:
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
```



In [65]:

```
function outer(X, Y, op)
    if (ndims(X)==1)
        X=reshape(X, (size(X,1), size(X,2)))
    end
    if (ndims(Y)==1)
        Y=reshape(Y, (size(Y,1), size(Y,2)))
    end
    Z=zeros(size(X)[1], size(Y)[2])
    for i in 1:size(X)[1], j in 1:size(Y)[2], k in size(X)[2]
        Z[i, j] = op(X[i,k], Y[k,j])
    end
    return Z
end
```

```
outer (generic function with 1 method)
```

In [66]:

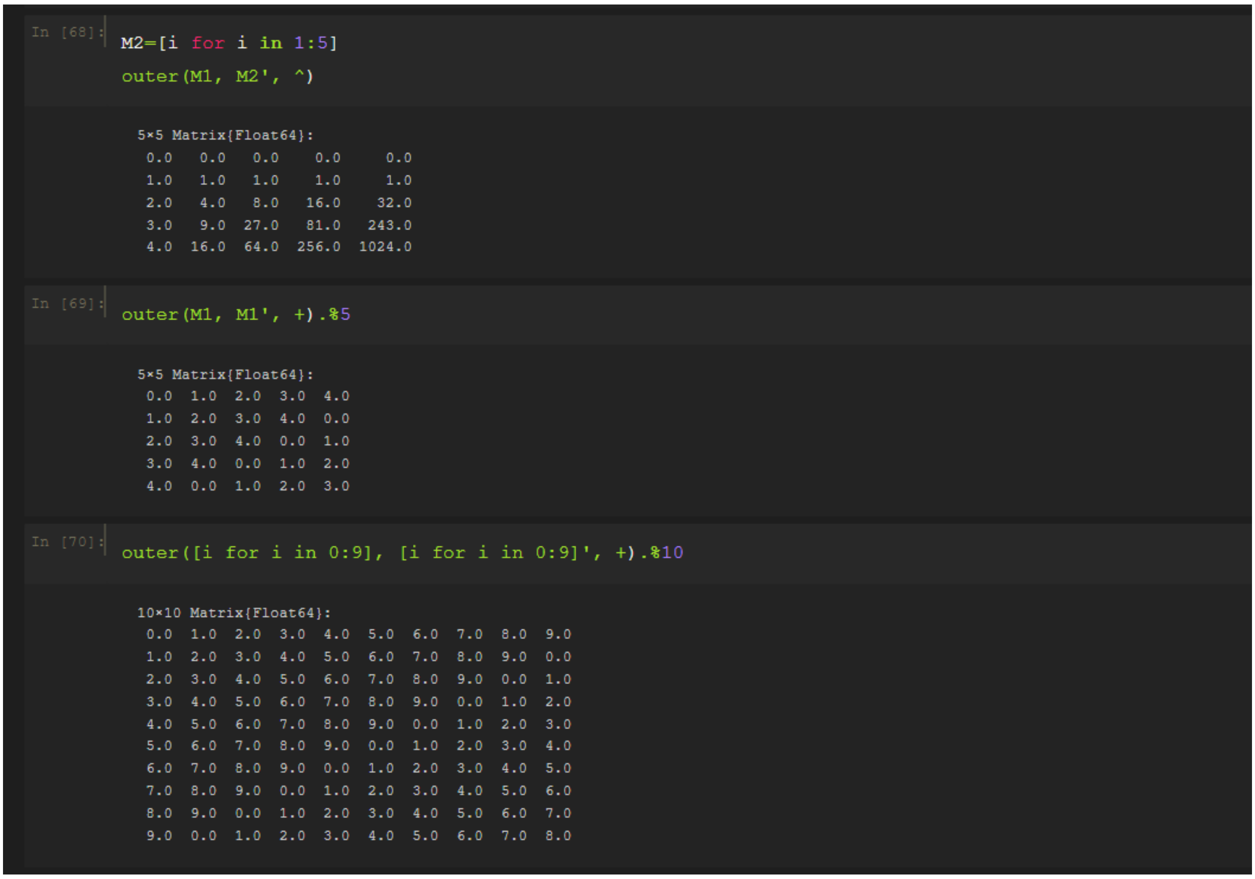
```
M1=[i for i in 0:4]
```

```
5-element Vector{Int64}:
 0
 1
 2
 3
 4
```

In [67]:

```
outer(M1, M1', +)
```

```
5×5 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0
 1.0  2.0  3.0  4.0  5.0
 2.0  3.0  4.0  5.0  6.0
 3.0  4.0  5.0  6.0  7.0
 4.0  5.0  6.0  7.0  8.0
```



```
In [68]: M2=[i for i in 1:5]
         outer(M1, M2', ^)
```

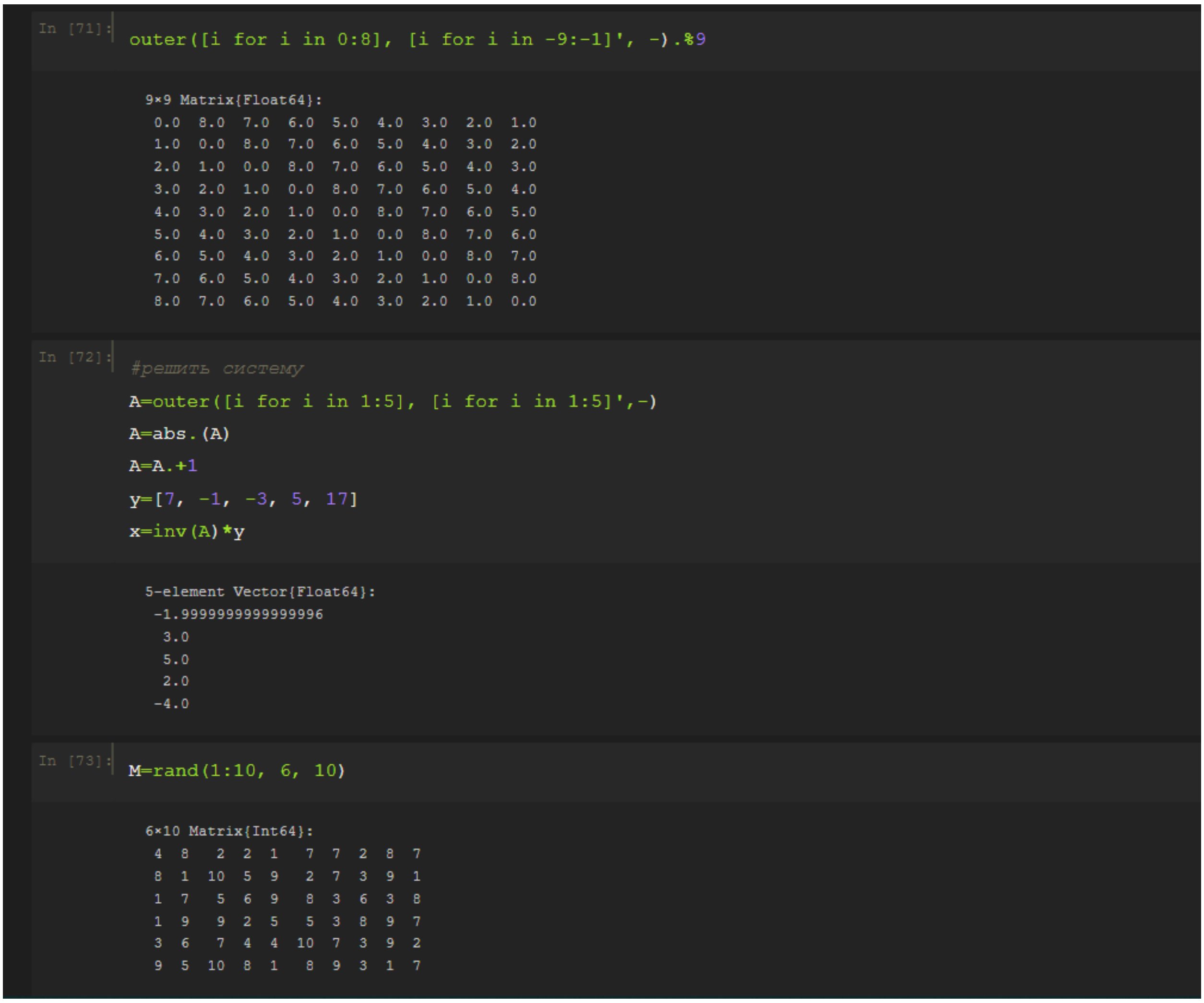
```
5×5 Matrix{Float64}:
 0.0   0.0   0.0    0.0     0.0
 1.0   1.0   1.0    1.0     1.0
 2.0   4.0   8.0   16.0    32.0
 3.0   9.0  27.0   81.0   243.0
 4.0  16.0  64.0  256.0  1024.0
```

```
In [69]: outer(M1, M1', +).%5
```

```
5×5 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0
 1.0  2.0  3.0  4.0  0.0
 2.0  3.0  4.0  0.0  1.0
 3.0  4.0  0.0  1.0  2.0
 4.0  0.0  1.0  2.0  3.0
```

```
In [70]: outer([i for i in 0:9], [i for i in 0:9]', +).%10
```

```
10×10 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0
 1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0
 2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0
 3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0
 4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0
 5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0
 6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0
 7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0
 8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0
 9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0
```



```
In [71]: outer([i for i in 0:8], [i for i in -9:-1]', -).%9
```

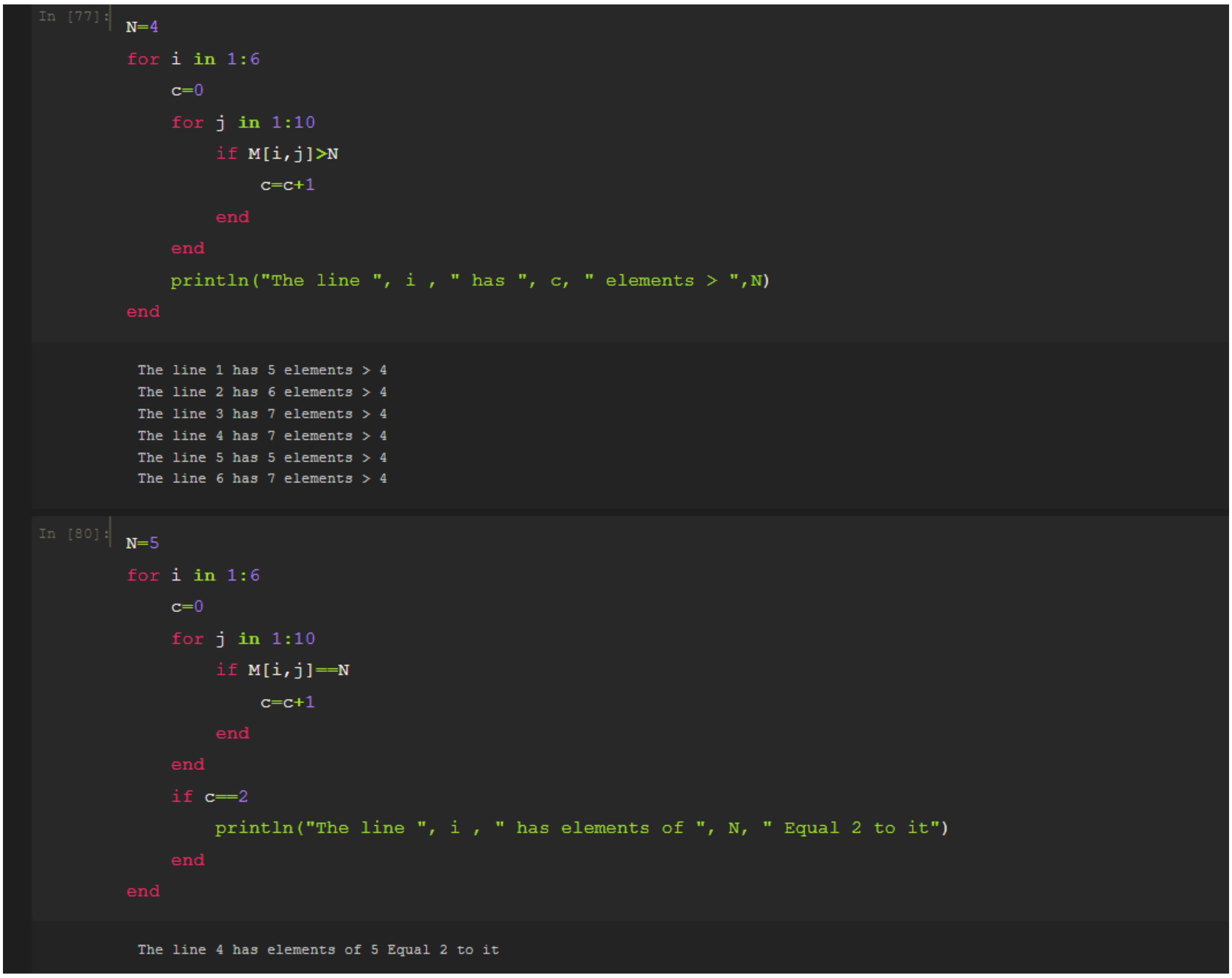
```
9×9 Matrix{Float64}:
 0.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0
 1.0  0.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0
 2.0  1.0  0.0  8.0  7.0  6.0  5.0  4.0  3.0
 3.0  2.0  1.0  0.0  8.0  7.0  6.0  5.0  4.0
 4.0  3.0  2.0  1.0  0.0  8.0  7.0  6.0  5.0
 5.0  4.0  3.0  2.0  1.0  0.0  8.0  7.0  6.0
 6.0  5.0  4.0  3.0  2.0  1.0  0.0  8.0  7.0
 7.0  6.0  5.0  4.0  3.0  2.0  1.0  0.0  8.0
 8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0  0.0
```

```
In [72]: #решить систему
         A=outer([i for i in 1:5], [i for i in 1:5]',-)
         A=abs.(A)
         A=A.+1
         y=[7, -1, -3, 5, 17]
         x=inv(A)*y
```

```
5-element Vector{Float64}:
 -1.9999999999999996
  3.0
  5.0
  2.0
 -4.0
```

```
In [73]: M=rand(1:10, 6, 10)
```

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



In [77]:

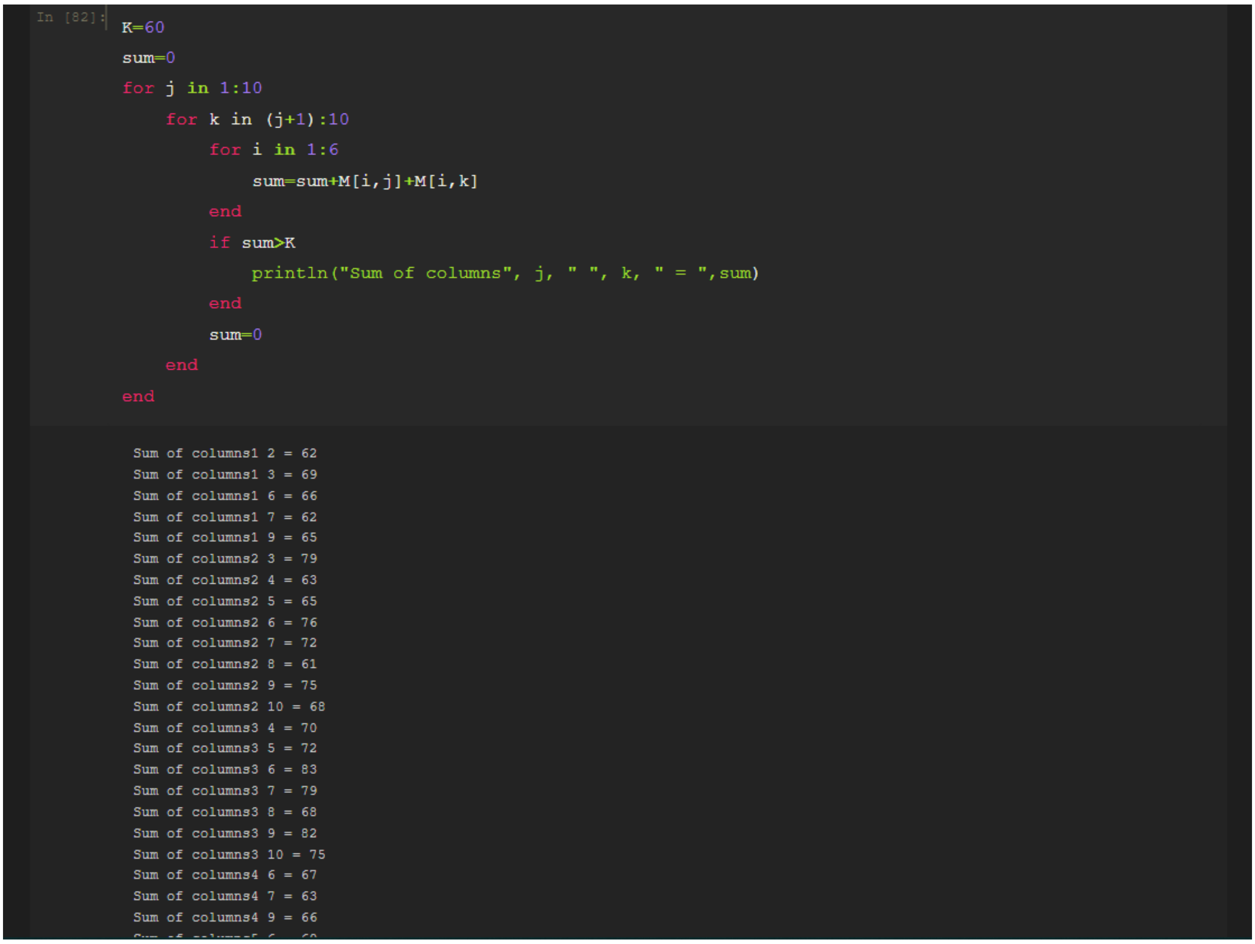
```
N=4
for i in 1:6
    c=0
    for j in 1:10
        if M[i,j]>N
            c=c+1
        end
    end
    println("The line ", i , " has ", c, " elements > ",N)
end
```

```
The line 1 has 5 elements > 4
The line 2 has 6 elements > 4
The line 3 has 7 elements > 4
The line 4 has 7 elements > 4
The line 5 has 5 elements > 4
The line 6 has 7 elements > 4
```

In [80]:

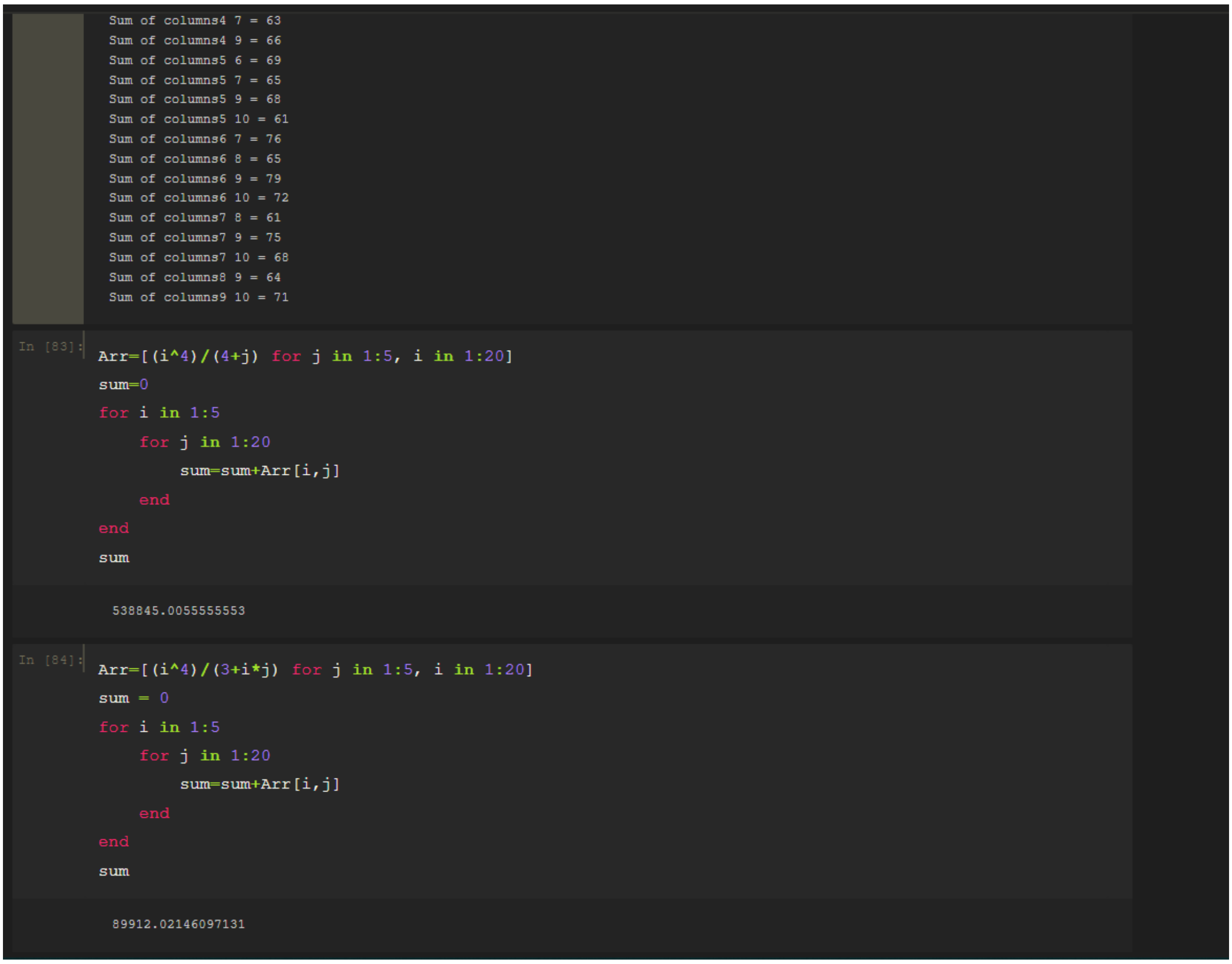
```
N=5
for i in 1:6
    c=0
    for j in 1:10
        if M[i,j]==N
            c=c+1
        end
    end
    if c==2
        println("The line ", i , " has elements of ", N, " Equal 2 to it")
    end
end
```

```
The line 4 has elements of 5 Equal 2 to it
```



```
In [82]: K=60
         sum=0
         for j in 1:10
             for k in (j+1):10
                 for i in 1:6
                     sum=sum+M[i,j]+M[i,k]
                 end
                 if sum>K
                     println("Sum of columns", j, " ", k, " = ",sum)
                 end
                 sum=0
             end
         end
```

```
Sum of columns1 2 = 62
Sum of columns1 3 = 69
Sum of columns1 6 = 66
Sum of columns1 7 = 62
Sum of columns1 9 = 65
Sum of columns2 3 = 79
Sum of columns2 4 = 63
Sum of columns2 5 = 65
Sum of columns2 6 = 76
Sum of columns2 7 = 72
Sum of columns2 8 = 61
Sum of columns2 9 = 75
Sum of columns2 10 = 68
Sum of columns3 4 = 70
Sum of columns3 5 = 72
Sum of columns3 6 = 83
Sum of columns3 7 = 79
Sum of columns3 8 = 68
Sum of columns3 9 = 82
Sum of columns3 10 = 75
Sum of columns4 6 = 67
Sum of columns4 7 = 63
Sum of columns4 9 = 66
```



```
Sum of columns4 7 = 63
Sum of columns4 9 = 66
Sum of columns5 6 = 69
Sum of columns5 7 = 65
Sum of columns5 9 = 68
Sum of columns5 10 = 61
Sum of columns6 7 = 76
Sum of columns6 8 = 65
Sum of columns6 9 = 79
Sum of columns6 10 = 72
Sum of columns7 8 = 61
Sum of columns7 9 = 75
Sum of columns7 10 = 68
Sum of columns8 9 = 64
Sum of columns9 10 = 71
```

```
In [83]: Arr=[(i^4)/(4+j) for j in 1:5, i in 1:20]
         sum=0
         for i in 1:5
             for j in 1:20
                 sum=sum+Arr[i,j]
             end
         end
         sum
```

```
538845.0055555553
```

```
In [84]: Arr=[(i^4)/(3+i*j) for j in 1:5, i in 1:20]
         sum = 0
         for i in 1:5
             for j in 1:20
                 sum=sum+Arr[i,j]
             end
         end
         sum
```

```
89912.02146097131
```

Выводы

Получены навыки работы с матрицами и функциями в Julia