

# **Лабораторная работа 2**

**Структуры данных**

Ланцова Яна Игоревна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Задание 1 . . . . .	8
3.2	Задание 2 . . . . .	9
3.3	Задание 3 . . . . .	9
3.4	Задание 4 . . . . .	12
3.5	Задание 5 . . . . .	13
3.6	Задание 6 . . . . .	13
<b>4</b>	<b>Выводы</b>	<b>15</b>

# Список иллюстраций

3.1	Примеры использования кортежей и словарей . . . . .	7
3.2	Примеры использования множеств . . . . .	8
3.3	Примеры использования массивов . . . . .	8
3.4	Работа с множествами . . . . .	9
3.5	Примеры операций над множествами элементов разных типов . .	9
3.6	Работа с массивами по заданиям 3.1-3.6 . . . . .	10
3.7	Работа с массивами по заданиям 3.7-3.10 . . . . .	10
3.8	Работа с массивами по заданию 3.11 . . . . .	10
3.9	Работа с массивами по заданиям 3.12-3.14 . . . . .	11
3.10	Работа с массивами по заданию 3.14 . . . . .	11
3.11	Работа с массивами по заданию 3.14 . . . . .	11
3.12	Работа с массивами по заданию 3.14 . . . . .	12
3.13	Работа с массивами по заданию 3.14 . . . . .	12
3.14	Работа с массивами по заданию 3.14 . . . . .	12
3.15	Создание массива квадратов . . . . .	13
3.16	Работа с пакетом Primes . . . . .	13
3.17	Вычисление сумм . . . . .	14

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

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

## 3 Выполнение лабораторной работы

Для начала выполним примеры из раздела про кортежи и словари(рис. 3.1). Кортеж (Tuple) – структура данных (контейнер) в виде неизменяемой индексируемой последовательности элементов какого-либо типа (элементы индексируются с единицы). В то время как словарь – неупорядоченный набор связанных между собой по ключу данных.

```
[1]: favoritelang = ("Python","Julia","R", "C++")
[1]: ("Python", "Julia", "R", "C++")
[2]: tuple_1 = (1, 2.0, "tmp", 53)
[2]: (1, 2.0, "tmp", 53)
[3]: tuple_2 = (a=2, b=1+2)
[3]: (a = 2, b = 3)
[5]: in("tmp", tuple_1), 0 in tuple_1
[5]: (true, false)
[6]: phonebook = Dict{"Иванов И.И." => ("867-5389", "333-5544"), "Бухгалтерия" => "555-2368"}
[6]: Dict{String, Any} with 2 entries:
      "Бухгалтерия" => "555-2368"
      "Иванов И.И." => ("867-5389", "333-5544")
[7]: keys(phonebook)
[7]: KeySet for a Dict{String, Any} with 2 entries. Keys:
      "Бухгалтерия"
      "Иванов И.И."
[8]: values(phonebook)
[8]: ValueIterator for a Dict{String, Any} with 2 entries. Values:
      "555-2368"
      ("867-5389", "333-5544")
[9]: haskey(phonebook, "Иванов И.И.")
[9]: true
```

Рис. 3.1: Примеры использования кортежей и словарей

Выполним примеры из раздела про множества(рис. 3.2). Множество, как структура данных в Julia, соответствует множеству, как математическому объекту, то есть является неупорядоченной совокупностью элементов какого-либо типа. Возможные операции над множествами: объединение, пересечение, разность; принадлежность элемента множеству.

```

[10]: A = Set([6, 99, 2, 47])
[10]: Set{Int64} with 4 elements:
      6
      99
      2
      47

[11]: B = Set(["rudn"])
[11]: Set{Char} with 4 elements:
      'r'
      'u'
      'd'
      'n'

[12]: first_set = Set([2, 3])
      second_set = Set([4, 5])
      issetequal(first_set, second_set)

[12]: false

[13]: first_and_second = union(first_set, second_set)
[13]: Set{Int64} with 4 elements:
      5
      4
      2
      3

[15]: intersect(first_set, second_set)

[15]: Set{Int64}()

[16]: setdiff(first_set, second_set)

[16]: Set{Int64} with 2 elements:
      2
      3

```

Рис. 3.2: Примеры использования множеств

Выполним примеры из раздела про массивы (рис. 3.3). Массив — коллекция упорядоченных элементов, размещённая в многомерной сетке. Векторы и матрицы являются частными случаями массивов.

```

[18]: a = [1, 0, 99]
      b = [4, 70, 8]

[18]: 3-element Vector{Int64}:
      4
      70
      8

[19]: c = rand(1,0)

[19]: 1x8 Matrix{Float64}:
      0.459876  0.475469  0.785781  0.00159815  - 0.311825  0.124381  0.263145

[21]: ones(5)

[21]: 5-element Vector{Float64}:
      1.0
      1.0
      1.0
      1.0
      1.0

[23]: c = transpose(b)

[23]: 1x3 transpose{::Vector{Int64}} with eltype Int64:
      4 70 8

[25]: ar = rand(10:20, 2, 5)

[25]: 2x5 Matrix{Int64}:
      13 19 20 12 16
      12 10 12 13 11

[26]: sort(ar, dims=1)

[26]: 2x5 Matrix{Int64}:
      12 10 12 12 11
      13 19 20 13 16

```

Рис. 3.3: Примеры использования массивов

Перейдем к выполнению заданий.

## 3.1 Задание 1

Даны множества:  $A = 0, 3, 4, 9$ ,  $B = 1, 3, 4, 7$ ,  $C = 0, 1, 2, 4, 7, 8, 9$ . Найдём  $P = A \cap B \cup A \cap B \cup A \cap C \cup B \cap C$  (рис. 3.4):



```

[27]: A = Set([0, 3, 4, 9])
      B = Set([1, 3, 4, 7])
      C = Set([0, 1, 2, 4, 7, 8, 9])

[27]: Set{Int64} with 7 elements:
      0
      4
      7
      2
      9
      8
      1

[29]: P = union(intersect(A, B), intersect(A, B), intersect(A, C), intersect(B, C))

[29]: Set{Int64} with 6 elements:
      0
      4
      7
      9
      3
      1

```

Рис. 3.4: Работа с множествами

## 3.2 Задание 2

Приведем свои примеры с выполнением операций над множествами элементов разных типов (рис. 3.5):

```

[32]: # объединение множеств:
      union(A, B)

[32]: Set{Int64} with 6 elements:
      0
      4
      7
      9
      3
      1

[33]: # разность множеств:
      setdiff(A, B)

[33]: Set{Int64} with 2 elements:
      0
      9

задание 3

[34]: length(C)

[34]: 7

[35]: 3 in C

[35]: false

```

Рис. 3.5: Примеры операций над множествами элементов разных типов

## 3.3 Задание 3

Создадим массивы разными способами с использованием циклов и встроенных функций (рис. 3.6 - 3.14):

```
[37]: arr1 = [i for i in 1:30];

[40]: arr1_1 = [1:30...];

[42]: println(arr1, "\n", arr1_1)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]

[44]: arr2 = reverse(arr1);
println(arr2)
[30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[48]: # объединим два массива
arr3 = vcat{1:30, (30:1):-1:1};
println(arr3)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 29, 28, 27, 26, 25, 24, 23,
22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[50]: tmp = [4, 6, 3]
println(tmp)
[4, 6, 3]

[51]: arr5 = fill(tmp[1], 10);
println(arr5)
[4, 4, 4, 4, 4, 4, 4, 4, 4, 4]

[52]: arr6 = fill(tmp, 10);
println(arr6)
[[4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3], [4, 6, 3]]
```

Рис. 3.6: Работа с массивами по заданиям 3.1-3.6

[illegible]

Рис. 3.7: Работа с массивами по заданиям 3.7-3.10

[illegible]

Рис. 3.8: Работа с массивами по заданию 3.11

```
[78]: vec12 = [2*i / i for i in 1:25];
println(vec12)

[2.0, 2.0, 2.6666666666666665, 4.0, 6.4, 10.666666666666666, 18.285714285714285, 32.0, 56.888888888888886, 102.4, 186.1818181818182, 341.333
333333333, 630.1538461538462, 1170.2857142857142, 2184.5333333333333, 4096.0, 7710.117647058823, 14563.555555555555, 27594.185263157893, 52
426.8, 99864.38095238095, 190650.18181818182, 364722.0809565217, 699850.6666666666, 1.342177286]
```

```
[81]: vec13 = ["fn1" for i in 1:30];
println(vec13)

["fn1", "fn2", "fn3", "fn4", "fn5", "fn6", "fn7", "fn8", "fn9", "fn10", "fn11", "fn12", "fn13", "fn14", "fn15", "fn16", "fn17", "fn18", "fn1
9", "fn20", "fn21", "fn22", "fn23", "fn24", "fn25", "fn26", "fn27", "fn28", "fn29", "fn30"]
```

```
[83]: x = rand(0:999, 250);
y = rand(0:999, 250);
```

```
[88]: vec14_1 = [y[i+1] - x[i] for i in 1:250-1];
println(vec13_1)

[-31, 131, -500, -143, -413, 184, -53, 631, 150, -470, 194, -645, -216, -765, 228, -530, -35, 39, 729, 87, 402, 766, 21, -779, -407, 504, -5
67, -645, -183, -27, 135, -546, 83, 737, 437, -441, -77, 610, -580, -566, 3, 254, -128, 13, -52, -830, 544, -263, 263, 234, 361, -493, -168,
-209, 400, -237, 245, 860, -445, -6, 857, -116, -76, -18, -460, -495, 299, 268, 445, 699, -420, 8, -115, -514, 304, 526, -173, 297, -362, -56,
96, 93, 27, 367, 196, 417, -129, -364, 89, 591, -920, -484, -103, -83, 278, 227, 439, -122, 629, 229, 354, 261, 208, 68, 235, 582, 275, -64,
-155, 372, -321, -102, -594, -20, -851, -41, -479, 100, -676, 737, -229, -820, -27, 494, 304, -214, 131, 123, 304, 627, 583, 521, 411, -806, 1
58, 171, 720, -545, -43, 707, -405, 690, 686, -850, 231, -385, -344, -22, 124, 37, -113, -591, -161, -327, 777, -647, -334, 828, -242, -111,
120, 811, 74, 63, 239, -250, 182, 154, -572, -442, -131, -118, 22, 672, 11, -359, 494, -608, 164, -304, -292, -333, 174, -451, 557, -307, 33
7, 510, -749, -814, -381, -154, 895, -15, 205, 316, -595, 685, 186, 490, 64, 42, 142, -45, 335, 153, -28, -252, 564, -146, -190, -436, -802,
-246, -11, 177, 29, -302, 457, 625, 422, 136, 261, 180, 340, 582, -493, 626, -27, 277, 451, 457, 220, -190, 62, 173, -731, -758, 457, 601,
709, -92, 278, 68, -23, -205, 651, 397, -116, 3]
```

```
[89]: vec14_2 = [x[i] + 2*x[i+1] - x[i+2] for i in 1:250-2];
println(vec13_2)

[1024, 2334, 799, 1160, -186, 1344, 156, 985, 1626, 090, 1552, 805, 1710, 1926, 1711, 935, 509, -153, 056, 399, -436, 1030, 1637, 2513, 174,
651, 1584, 2161, 500, -90, 1278, 1770, 627, 369, 913, 2405, 408, 1060, 1823, 1752, 651, 1376, 2019, 1398, 2183, 934, 1182, 611, 164, 070, 1
417, 1929, 837, 944, 1290, 1416, -201, 1479, 1612, 222, 932, 440, -47, 1571, 2083, 1647, 560, 939, 344, 1139, 1283, 980, 2251, 1353, 179, 12
38, 1315, 1523, 1964, 1314, 2116, 1591, 204, 230, 381, 975, 424, -836, 1326, 1401, 1774, 377, 910, 074, 072, 1148, 332, 476, 431, 519, 242,
103, 1121, 302, 344, 390, 1963, 686, 1247, 603, 1103, 1625, 2476, 627, 1024, 944, 2324, 700, 41, 1274, 2185, 942, -619, 1305, 1429, 2000, 11
00, 174, -329, 1187, 510, 1865, 681, 1802, 115, 426, 2340, 347, 1850, 1040, -515, 1284, 1510, 2083, 1784, 20, 1144, 542, 241, 1444, 756, 235
6, 197, 1083, 2692, 730, 817, 569, 1395, 449, 271, 110, 116, 892, 446, 507, 1236, 2064, 481, 789, 1475, 825, -140, 1634, 340, 1370, 908, 127
```

Рис. 3.9: Работа с массивами по заданиям 3.12-3.14

```
[90]: vec14_3 = [sin(y[i])/cos(x[i+1]) for i in 1:250-1];
println(vec14_3)

[0.7473890761102641, 1.3080914156295154, 4.599373499821138, -1.2400887358372137, -0.3566014197704019, -5.9087063246768645, -0.89724985355929
48, 0.8511732921191558, -2.1156099517838056, 14.1077569299659, 0.3191680436602564, -0.766863944094296, 0.10302303143715785, -0.60817547279
94285, -1.506102499929006, 0.0824239630827240, 1.0011008077001183, 0.302187011766545, -0.7592741917046908, 0.0550529109223750, 0.6417008328
135184, 0.0798474491088568, -0.31217595859127695, 14.008459081016229, -3.0923579084309636, 0.10702770680635906, -0.5088829453732117, -1.0005
334908144483, 0.6768157283168107, -1.0466712188519456, 1.3421572418870846, -0.6055699644024115, -2.673308178247664, 4.871633694203908, -0.795
32761340734071, -18.153110638613946, -0.5172144176703048, 0.4017151265335428, -3.8432602777083013, 0.8397707609313448, 1.66432632320120536,
4.845901741871703, -1.120625164057577, -1.0673210204737072, -0.00190656308173133, 0.1559930296012099, -0.7327690087141722, 0.23603608073571
328, 1.2458716095464781, -0.9038382760117768, 0.8537890131682548, -0.7837639195346212, -0.8291926863127305, 1.0010658501286343, -0.859745678
5034667, 1.3135113746588082, 2.169801906609702, 0.734077425140043, -0.9084112335599723, -0.5286891783796658, 0.03836140025085579, 0.117515158
335359076, 0.6322839067314325, 0.7338497581375089, -0.3040952680640895, 0.005081648282896, 9.706057482228735, 0.11456969302781828, 0.409017
505590942706, -0.8209364253549514, -24.777312422859408, -2.2540747796053027, -0.129406332677007, 0.1217070664932518, -2.150231746007901,
1.031235855722456, 0.1253023628166324, 0.4534253267163959, 0.52318444383447154, 0.0845756200107418, -0.0271840538089779618, 0.273195681398424
5, -0.708371568543003, 0.7955001406101527, -0.4624831554839136, -1.6795890210114278, -0.0095018411347278, 0.6323258473043197, 1.1423086624
34277, 0.3600721551353475, -1.0715914969727000, 0.9010070074552461, -1.1231607510709631, -1.7637708544700142, -0.710047396326540, -0.1061
485416293908, -3.64189032226735, 0.42073643065214333, -1.034301530533087, 0.08941070620062127, -5.125285895004963, -1.002512263560631, 0.0
474004715326545, 1.456258311555779, -1.2891764606309002, -1.0274735749675783, -0.6437185521307713, 0.6796179236971972, -0.553170435228858,
-3.0025558930378225, 1.3587478501960284, -0.23509675735730173, 0.10950840897906366, -10.3464820294, -0.6180820652100012, 1.0163651232396411,
-3.531331618343402, 1.306702842650270, -1.0335506595950608, -0.2368335507204167, 0.1909061367390097, 0.0, 0.6506400112190067, 1.7111333942
22655, -0.094529632673156, 0.5800772710662155, 0.2088724089447745, 0.009047334007058088, 3.9595381780593755, -7.241383523048007, 1.8375247353
510624, -1.3774860422830657, -1.1655027686854833, 0.30105977605935325, -1.0080341239462571, -1.4584181394675446, 0.1053215460059022, 0.70029
37900383628, 0.7115700372613735, 1.7804993169575731, 0.8205905797271945, 0.8756857612452192, 3.7629216713208553, 1.8348166595062872, -0.2030
405105770602, 3.1405957107704593, 2.730007097451709, 1.0505055469632172, -0.4241714059274575, -1.1232452200521796, 0.853613159979108, -2.5
082080126901156, -0.9618021223380751, 1.1130207927377194, 0.8111731040684310, 0.34017181035740707, 1.2909043087850016, -1.4210353136734407, -1
2707239510881616, 1.3103303694071806, -0.409032566037324, 0.8161652471110976, 1.3056506979033837, -0.3431248107421011, 0.50742221032824,
0.7374084839733527, -29.073476810961207, -1.0125062645507008, 2.7839175237873954, -1.051329799766767, 0.3163645068282446, 1.992065703930372
9, 0.5040000130183655, -1.2220024053451195, 1.070959647411954, -1.2292050017300004, -2.0610347442276455, -2.112740263446209, -0.92453907617
70192, -1.0978308166919083, -35.67270186071706, -1.6544427376302204, -0.33206216457649254, -0.4183616423465040, 0.7607543185066473, 0.99654
5347904983, 1.1322084133226294, -0.807549117836715, -0.9208534901213675, 0.0095742416443351, -1.9970976837709980, -0.03076466537155308, -1.
00970070141385712, -0.1792385183034481, -0.9729923573736801, -0.40337014115347656, 0.9944390598528742, -0.4014003891858844, -0.28999330406086
20, 0.3912025146642029, -0.7089935213182223, -2.356251000807717, 0.19495615097302713, 0.13350669540556137, -0.6709647284934092, 1.4041530131
```

Рис. 3.10: Работа с массивами по заданию 3.14

```
[91]: sum14_4 = sum(exp(-x[i+1]) / (x[i] + 10) for i in 1:250-1)

[91]: 1.6198145953442972e-7
```

```
[94]: y_bigger_000 = y[y .> 600];
y_bigger_indices = findall(y .> 600)
println(y_bigger_000, "\n", y_bigger_indices)
```

```
[739, 770, 718, 665, 730, 934, 692, 998, 777, 936, 863, 689, 962, 866, 867, 816, 686, 696, 780, 911, 641, 826, 995, 994, 717, 972, 996, 842,
934, 991, 619, 655, 842, 631, 983, 760, 932, 728, 646, 625, 849, 603, 888, 734, 722, 668, 632, 723, 623, 843, 681, 826, 725, 633, 726, 682,
930, 696, 652, 740, 765, 938, 729, 800, 874, 827, 830, 971, 741, 652, 868, 602, 972, 675, 841, 655, 678, 871, 683, 653, 862, 959, 776, 972,
815, 873, 873, 638, 635, 727, 605, 783, 839, 600, 776, 810, 860, 638, 731, 711, 843, 806, 885, 817, 769, 862, 801, 636]
```

```
[1, 2, 3, 0, 9, 10, 12, 16, 20, 23, 24, 34, 35, 36, 30, 42, 43, 44, 45, 46, 40, 52, 56, 50, 59, 62, 60, 70, 71, 73, 76, 77, 70, 79, 82,
83, 84, 86, 90, 95, 96, 97, 99, 105, 106, 107, 109, 110, 114, 118, 120, 123, 124, 126, 127, 128, 130, 131, 132, 133, 136, 137, 140, 14
2, 143, 145, 149, 154, 155, 157, 158, 161, 162, 168, 173, 174, 177, 179, 107, 180, 182, 183, 189, 192, 193, 195, 196, 198, 199, 200, 202, 204, 205, 207, 20
8, 209, 220, 221, 223, 224, 228, 229, 232, 233, 236, 240, 241, 247, 250]
```

```
[96]: x_corres = x[y_bigger_indices];
```

```
[97]: for i in 1:length(y_bigger_indices)
idx = y_bigger_indices[i]
println("x[idx]=x[x[idx]], y=y[y[idx]]")
end
```

```
1: x=009, y=739
2: x=507, y=778
3: x=959, y=718
4: x=107, y=665
9: x=776, y=730
10: x=674, y=934
12: x=700, y=692
16: x=601, y=998
20: x=437, y=777
23: x=042, y=936
24: x=024, y=803
24: x=225, y=689
35: x=429, y=962
36: x=714, y=866
```

Рис. 3.11: Работа с массивами по заданию 3.14

```
[100]: vec14_7 = [abs(x[i] - mean(x))**0.5 for i in 1:250];
       println(vec14_7)

[17.830872104302696, 9.794896630880704, 21.631920857843394, 17.890220792377047, 3.307567081708245, 21.818799233688367, 15.064527871792866, 1
9.59744881355734, 16.880165875962238, 13.525531412850292, 2.634387974463898, 16.998235202514408, 3.8652296180175374, 22.2697103708615332, 16.
701496938897424, 10.485227783774488, 15.167728900538089, 17.40861855518697, 21.048990474604715, 7.35255857786072, 20.616985329040166, 17.91814
722554080, 18.73339264522046, 18.246643526061336, 19.024720768105783, 21.77291895911005, 14.897650821522163, 17.347622315464445, 14.80765802
1522163, 20.834106652314134, 13.155227898001767, 17.086251783232036, 10.721007415350481, 16.311345744603663, 7.0778169564924525, 14.93117543
9328278, 21.28238708415952, 17.140177745754793, 18.9457119158092737, 18.3559254738080170, 18.054241227869537, 7.004284403134904, 20.6140728629
73973, 20.172753902231594, 14.210559454152394, 19.0738097393209747, 14.458907289280196, 4.993996395673509, 18.948870594787608, 17.94603820169
0046, 11.955751753863074, 16.61144183980853, 16.122654068220125, 12.290646080349253, 10.145935146640632, 9.640539483996023, 4.365775994806
15, 19.46946326943812, 20.147952749597167, 8.003749121505496, 18.763261976532757, 2.839013913315678, 17.834236737242218, 10.102474944289641,
22.515328112199477, 17.978531433432903, 7.2069410986909, 15.429193109168088, 1.4352700094407331, 14.1088624638053, 15.42530886795877, 10.93117543
9328278, 13.301879566437218, 21.81604913819182, 11.83469475736489, 13.231024147812596, 17.69011023142592, 13.962091533064115, 19.23362
4372703417, 19.8470210056047, 13.264237633576006, 20.34551547630394, 11.404385121522335, 21.332135383831865, 16.108067356329300, 15.068510
211696443, 10.726602444390302, 21.21461760202149, 21.378961621182633, 21.305867736377227, 10.34117981663681, 12.997692102831179, 18.62956462
4966082, 8.940917178902845, 10.726602444390302, 6.485368146836385, 9.437160589923222, 19.648409604850976, 14.527904184706005, 17.37987341726
055, 16.24992307674101, 19.158009893921234, 16.86424601485245, 2.014944167908989, 20.126102454275642, 11.578428210026033, 5.921148537234905,
13.696192525460347, 15.403069755487168, 14.200756202494286, 9.851003369400251, 17.280881433948587, 19.492049661136285, 20.304607130317564, 1
6.794642002734086, 14.385409274678285, 9.483670175622937, 20.614072862973973, 20.051433863941003, 10.249878048055011, 18.1367031182627, 16.1
53637361288013, 18.763261976532757, 20.07635425070996, 21.046139788569302, 12.646738710039044, 17.774701122663075, 17.494570586327633, 20.54
409090941922, 18.49406415197473, 14.315725619053637, 11.707262703125702, 20.07336543781336, 14.630709452304311, 16.611441839808553, 21.95506
482013405, 13.37684566704722, 20.78316626503286, 10.001666589513317, 20.46802384208109, 18.816482136680065, 18.629564624966082, 21.180284265
581355, 15.777832550765647, 20.46802384208109, 3.7496666518505353, 20.27461466958127, 11.222299229658779, 17.918147225640808, 8.7212384441660
```

Рис. 3.12: Работа с массивами по заданию 3.14

```
[103]: y_close_max = y[abs.(y .- maximum(y)) .<= 200];
       println(length(y_close_max))
       50

[107]: x_even = x[x .% 2 .== 0]
       x_odd = x[x .% 2 .== 1]
       println(length(x_even), " ", length(x_odd))
       123 127

[108]: x_seven = x[x .% 7 .== 0]
       println(length(x_seven))
       32

[109]: println(sortperm(x))

[752, 41, 590, 582, 277, 310, 967, 853, 632, 282, 209, 617, 874, 571, 525, 261, 15, 740, 610, 309, 660, 65, 330, 923, 34, 594, 00, 506, 376,
792, 713, 45, 937, 238, 820, 690, 489, 551, 867, 577, 718, 240, 730, 498, 555, 415, 992, 195, 581, 770, 610, 180, 48, 489, 229, 122, 606, 0
9, 91, 797, 315, 132, 25, 820, 683, 633, 944, 751, 105, 999, 871, 667, 987, 188, 790, 487, 759, 611, 181, 936, 543, 459, 979, 402, 85, 233,
472, 64, 173, 602, 998, 394, 40, 879, 169, 644, 634, 653, 139, 501, 767, 109, 668, 410, 17, 171, 427, 783, 351, 15, 230, 934, 451, 124, 69,
885, 137, 384, 189, 456, 349, 713, 489, 866, 160, 66, 477, 462, 57, 814, 227, 170, 967, 930, 144, 770, 185, 368, 340, 709, 685, 824, 144, 44
9, 660, 695, 945, 686, 456, 88, 521, 746, 145, 543, 886, 264, 696, 01, 825, 316, 743, 212, 107, 357, 30, 199, 916, 807, 835, 448, 442, 225,
780, 916, 149, 870, 112, 959, 651, 86, 251, 139, 59, 36, 670, 112, 280, 776, 809, 170, 354, 905, 894, 153, 77, 514, 437, 587, 25, 898, 657,
438, 557, 850, 790, 516, 386, 144, 940, 912, 166, 253, 804, 907, 89, 376, 146, 449, 241, 824, 714, 197, 964, 229, 806, 452, 910, 167, 109, 3
18, 402, 693, 105, 361, 674, 292, 842, 9, 706, 429, 910, 897, 494, 106, 861, 729, 584, 767, 483, 601]

[110]: reverse(last(sort(x), 10))

[110]: 10-element Vector{Int64}:
       999
       998
       992
       907
       979
       967
       967
       864
       959
```

Рис. 3.13: Работа с массивами по заданию 3.14

```
--
[114]: vec_uni_x = unique(x);
       println(vec_uni_x)

[009, 587, 959, 171, 502, 15, 718, 107, 776, 674, 490, 700, 506, 087, 770, 601, 261, 180, 40, 437, 66, 170, 042, 824, 853, 17, 713, 792, 57,
318, 783, 606, 225, 429, 714, 944, 197, 850, 828, 683, 442, 916, 898, 693, 886, 282, 516, 132, 169, 634, 767, 751, 340, 594, 584, 472, 112,
897, 427, 139, 483, 173, 389, 998, 814, 543, 253, 489, 292, 729, 611, 668, 967, 351, 316, 804, 686, 861, 885, 667, 905, 361, 36, 229, 264, 3
76, 41, 34, 945, 598, 660, 144, 571, 449, 402, 105, 280, 189, 227, 124, 233, 487, 86, 357, 456, 879, 251, 695, 394, 790, 871, 907, 209, 698,
501, 09, 306, 820, 752, 08, 934, 651, 807, 105, 69, 149, 696, 354, 804, 277, 9, 670, 923, 167, 910, 137, 940, 740, 477, 80, 617, 415, 759, 9
79, 91, 964, 936, 494, 315, 555, 30, 166, 230, 451, 248, 501, 743, 181, 632, 656, 199, 797, 835, 825, 866, 770, 410, 709, 25, 368, 525, 786,
937, 930, 77, 106, 610, 557, 452, 145, 521, 685, 59, 730, 448, 867, 912, 212, 514, 602, 653, 992, 577, 40, 195, 618, 122, 438, 146, 999, 85,
870, 64, 349, 657, 338, 644, 874, 65, 168, 153, 241, 45, 459, 462, 230, 109, 384, 633, 746]
```

Рис. 3.14: Работа с массивами по заданию 3.14

## 3.4 Задание 4

Создадим массив `squares`, в котором будут храниться квадраты всех целых чисел от 1 до 100 (рис. 3.15):

```

[116]: vec15 = [i^2 for i in 1:100];
println(vec15)
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225, 1296, 1369, 1444, 1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304, 2401, 2500, 2601, 2704, 2809, 2916, 3025, 3136, 3249, 3364, 3481, 3600, 3721, 3844, 3969, 4096, 4225, 4356, 4489, 4624, 4761, 4900, 5041, 5184, 5329, 5476, 5625, 5776, 5929, 6084, 6241, 6400, 6561, 6724, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000]

```

Рис. 3.15: Создание массива квадратов

## 3.5 Задание 5

Подключим пакет Primes (функции для вычисления простых чисел). Сгенерируем массив `myprimes`, в котором будут храниться первые 168 простых чисел. Определим 89-е наименьшее простое число. Получим срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа (рис. 3.16).

```

[119]: using Pkg
      Pkg.add("Primes")

      Updating registry at '~/.julia/registries/General.toml'
      Resolving package versions...
      Installed IntegerMathUtils - v0.1.3
      Installed Primes - v0.5.7
      Updating '~/.julia/environments/v1.11/Project.toml'
      [27ebfcd6] + Primes v0.5.7
      Updating '~/.julia/environments/v1.11/Manifest.toml'
      [19e54d08] + IntegerMathUtils v0.1.3
      [27ebfcd6] + Primes v0.5.7
      Precompiling project...
      968.7 ms ✓ IntegerMathUtils
      907.2 ms ✓ Primes
      2 dependencies successfully precompiled in 3 seconds. 39 already precompiled.

[123]: using Primes
      myprimes = primes(prime(168));
      println(myprimes)

      [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]

[124]: myprimes[89]

[124]: 461

[126]: println(myprimes[89:99])

      [461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523]

```

Рис. 3.16: Работа с пакетом Primes

## 3.6 Задание 6

Вычислим следующие выражения (рис. 3.17).

```
[127]: sum6_1 = sum(1^3 + 41^2 for i in 10:100)
[127]: 26852735
[128]: sum6_2 = sum(2^1/i + 3^1/i^2 for i in 1:25)
[128]: 2.1291704368143802e9
[130]: mys_sum = 1
      num = 1
      for i in 2:2:38
          num += 1/(i+1)
          mys_sum += num
      end
      print(mys_sum)
6.976346137897618
```

Рис. 3.17: Вычисление сумм

## 4 Выводы

В результате выполнения данной лабораторной работы я изучила несколько структур данных, реализованных в Julia, научилась применять их и операции над ними для решения задач.