

Лабораторная работа №2.

Евдокимов Иван Андреевич. НФИбд-01-20

15 ноября, 2023, Москва, Россия

Российский Университет Дружбы Народов

Цель лабораторной работы

Цель лабораторной работы

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

Процесс выполнения лабораторной работы

Пункт 1

1. Даны множества:

$$A = 0, 3, 4, 9, B = 1, 3, 4, 7, C = 0, 1, 2, 4, 7, 8, 9.$$

Найти $P = A \cap B \cup A \cap C \cup B \cap C$

```
# Заданные множества
A = Set([0, 3, 4, 9])
B = Set([1, 3, 4, 7])
C = Set([0, 1, 2, 4, 7, 8, 9])

# Находим пересечения множеств
intersection_AB = intersect(A, B)
intersection_AC = intersect(A, C)
intersection_BC = intersect(B, C)

# Объединяю результаты
P = union(intersection_AB, intersection_AC, intersection_BC)

# Вывожу результат
println("Множество P: ", P)

Множество P: Set([0, 4, 7, 9, 3, 1])
```

Рис. 1: Номер 1

2. Привожу свои примеры с выполнением операций над множествами элементов разных типов.

Пример 1: Множество целых чисел и множество строк

```
# Создаю множество
integer_set = Set([1, 2, 3, 4, 5])
string_set = Set(["apple", "banana", "orange"])

# Пересечение множеств
intersection_set = intersect(integer_set, string_set)
println("Пересечение: ", intersection_set)

# Объединение множеств
union_set = union(integer_set, string_set)
println("Объединение: ", union_set)

Пересечение: Set{Any}()
Объединение: Set{Any}[5, 4, 2, "orange", "banana", 3, 1, "apple"]
```

Рис. 2: Пример 1: Множество целых чисел и множество строк

2.2. Пример 2: Множество дробных чисел и множество СИМВОЛОВ

```
# Создай множества
float_set = Set([1.1, 2.2, 3.3, 4.4, 5.5])
char_set = Set(['a', 'b', 'c'])

# Пересечение множеств
intersection_set = intersect(float_set, char_set)
println("Пересечение: ", intersection_set)

# Объединение множеств
union_set = union(float_set, char_set)
println("Объединение: ", union_set)

Пересечение: Set{Any}{}
Объединение: Set{Any}[2.2, 1.1, 'a', 'c', 3.3, 4.4, 5.5, 'b']
```

Рис. 3: Пример 2: Множество дробных чисел и множество СИМВОЛОВ

2.3. Пример 3: Множество булевых значений и множество кортежей

```
# Создаю множества
bool_set = Set([true, false])
tuple_set = Set([(1, "apple"), (2, "banana"), (3, "orange")])

# Пересечение множеств
intersection_set = intersect(bool_set, tuple_set)
println("Пересечение: ", intersection_set)

# Объединение множеств
union_set = union(bool_set, tuple_set)
println("Объединение: ", union_set)

Пересечение: Set{Any}{}
Объединение: Set{Any[false, (1, "apple"), (2, "banana"), true, (3, "orange")]}
```

Рис. 4: Пример 3: Множество булевых значений и множество кортежей

3. Создаю разными способами:

3.1 массив $(1, 2, 3, \dots, N - 1, N)$, N выберите больше 20 ;

```
N = 25 # Замените это значение на любое число больше 20

arr = collect(1:N) # Создание массива от 1 до N
println(arr)

[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]
```

Рис. 5: 3.1(способ 1)

3.1.(способ 2):

3.1 массив $(1, 2, 3, \dots, N - 1, N)$, N выберите больше 20 ;

```
N = 25 # Замените это значение на любое число больше 20

arr = 1:N # Создание массива от 1 до N
arr = collect(arr) # Преобразование в лямбда-массив, если нужно
println(arr)
```

[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]

Рис. 6: 3.1(способ 2)

3.2.(способ 1):

3.2 массив $(N, N - 1 \dots, 2, 1)$, N выберите больше 20 ;

```
N = 25 # Замените это значение на любое число больше 20

arr = collect(N:-1:1) # Создание массива от N до 1
println(arr)

[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]
```

Рис. 7: 3.2(способ 1)

3.2.(способ 2):

3.2 массив $(N, N - 1 \dots, 2, 1)$, N выберите больше 20 ;

```
N = 25 # Замените это значение на любое число больше 20  
  
arr = reverse([i for i in range(1, N)]) # Создание массива от N до 1 и затем его обращение  
print(arr)
```

[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]

Рис. 8: 3.2(способ 2)

3.3.(способ 1):

3.3 массив $(1, 2, 3, \dots, N - 1, N, N - 1, \dots, 2, 1), N$
выберите больше 20 ;

```
N = 25 # Замените это значение на любое число больше 20

arr = vcat(1:N, reverse(1:N-1)) # Создание массива с помощью vcat и reverse
println(arr)
```

[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, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,

Рис. 9: 3.3(способ 1)

3.3.(способ 2):

3.3 массив $(1, 2, 3, \dots, N - 1, N, N - 1, \dots, 2, 1), N$
выберите больше 20 ;

```
N = 25 # Замените это значение на любое число больше 20

arr = vcat(1:N, reverse(1:N-1)) # Создание массива с помощью vcat и reverse
println(arr)

[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, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,
```

Рис. 10: 3.3(способ 2)

3.4.(способ 1):

3.4 массив с именем tmp вида (4, 6, 3);

```
tmp = [4, 6, 3]  
println(tmp)  
[4, 6, 3]
```

Рис. 11: 3.4(способ 1)

3.4.(способ 2):

3.4 массив с именем tmp вида (4, 6, 3);

```
tmp = Int[]  
push!(tmp, 4)  
push!(tmp, 6)  
push!(tmp, 3)  
println(tmp)
```

```
[4, 6, 3]
```

Рис. 12: 3.4(способ 2)

3.5.(способ 1):

3.5 массив, в котором первый элемент массива tmp повторяется 10 раз;

```
tmp = [4, 6, 3]
first_element = tmp[1]
arr1 = [first_element for _ in range(10)]
print(arr1)
```

```
[4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
```

Рис. 13: 3.5(способ 1)

3.5.(способ 2):

3.5 массив, в котором первый элемент массива tmp повторяется 10 раз;

```
tmp = [4, 6, 3]
first_element = tmp[1]
arr2 = fill(first_element, 10)
println(arr2)
```

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4]

Рис. 14: 3.5(способ 2)

3.6.(способ 1):

3.6 массив, в котором все элементы массива tmp повторяются 10 раз;

```
tmp = [4, 6, 3]
arri_repeated = [elem for elem in tmp for _ in range(10)]
println(arri_repeated)
```

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

Рис. 15: 3.6(способ 1)

3.6.(способ 2):

3.6 массив, в котором все элементы массива tmp повторяются 10 раз;

```
tmp = [4, 6, 3]
arr2_repeated = repeat(tmp, inner=10)
println(arr2_repeated)
```

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

Рис. 16: 3.6(способ 2)

3.7.(способ 1):

3.7 массив, в котором первый элемент массива tmp встречается 11 раз, второй элемент - 10 раз, третий элемент - 10 раз;

```
tmp = [4, 6, 3] # Замените это значение на ваши элементы
tmp_repeated = [if i < 12 tmp[1] elseif i < 22 tmp[2] else tmp[3] end for i in 1:31]
println(tmp_repeated)

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3]
```

Рис. 17: 3.7(способ 1)

3.7.(способ 2):

3.7 массив, в котором первый элемент массива tmp встречается 11 раз, второй элемент - 10 раз, третий элемент - 10 раз;

```
tmp = [4, 6, 3] # Замените это значение на ваши элементы
tmp_repeated = [fill(tmp[1], 11); fill(tmp[2], 10); fill(tmp[3], 10)]
println(tmp_repeated)

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3]
```

Рис. 18: 3.7(способ 2)

3.8.(способ 1):

3.8 массив, в котором первый элемент массива tmp встречается 10 раз подряд, второй элемент 20 раз подряд, третий элемент 30 раз подряд;

```
tmp = [4, 6, 3] # Замените это значение на ваши элементы
tmp_repeated = [if i < 11 tmp[1] elseif i < 31 tmp[2] else tmp[3] end for i in 1:61]
println(tmp_repeated)
```

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3]

Рис. 19: 3.8(способ 1)

3.9.(способ 1):

3.9 массив из элементов вида $2^{tmp[i]}$, $i = 1, 2, 3$, где элемент $2^{tmp[3]}$ встречается 4 раза; посчитайте в полученном векторе, сколько раз встречается цифра 6, и выведите это значение на экран;

```
# Заданный массив tmp
tmp = [4, 6, 3]

# Создание массива с использованием генератора
arr1 = [2**(tmp[i]) for i in 1:3]

# Объединение массивов в один
arr_combined = vcat(arr1[1], arr1[2], arr1[3], arr1[3], arr1[3], arr1[3])

println(arr_combined)

# Подсчет количества цифры 6 в полученном векторе
count_of_six = count(x-> contains(string(x), "6"), arr_combined)

# Вывод результата на экран
println("Количество цифры 6 в векторе: ", count_of_six)

[16, 64, 8, 8, 8, 8]
Количество цифры 6 в векторе: 2
```

Рис. 21: 3.9(способ 1)

3.9.(способ 2):

3.9 массив из элементов вида $2^{tmp[i]}$, $i = 1, 2, 3$, где элемент $2^{tmp[3]}$ встречается 4 раза; посчитайте в полученном векторе, сколько раз встречается цифра 6, и выведите это значение на экран;

```
# Заданный массив tmp
tmp = [4, 6, 3]

# Второй способ создания массива с использованием цикла
arr2 = zeros(Int, 3)
for i in 1:3
    arr2[i] = 2^(tmp[i])
end

# Объединение массивов в один
arr_combined = vcat(arr2[1], arr2[2], arr2[3], arr2[3], arr2[3], arr2[3])

println(arr_combined)

# Подсчет количества цифры 6 в полученном векторе
count_of_six = count(x-> occursin("6", string(x)), arr_combined)

# Вывод результата на экран
println("Количество цифры 6 в векторе: ", count_of_six)

[16, 64, 8, 8, 8, 8]
Количество цифры 6 в векторе: 2
```

Рис. 22: 3.9(способ 2)

3.10.(способ 1):

3.10 вектор значений $y = e^x \cos(x)$ в точках $x = 3, 3.1, 3.2, \dots, 6$, найдите среднее значение y ;

```
using Statistics

# Создание вектора x
x_values = 3:0.1:6

# Вычисление значений функции y=e^x * cos(x)
y_values = [exp(x) * cos(x) for x in x_values]

# Нахождение среднего значения y
mean_y = mean(y_values)

# Вывод результатов
println("Значения y: ", y_values)
println("Среднее значение y: ", mean_y)
```

Значения y: [-19.884530844146987, -22.178753389342127, -24.490696732001293, -26.77318244299338, -28.969237768093574, -31.011186439374516, -32.819774760338504, -34.35719361853035, -35.86283371230767, -35.68773248011913, -34.68504225166807, -32.693695428321746, -29.538816297262983, -25.032529229039966, -18.975233154958957, -11.3620985182057503, 10.632038010191998, 25.046704998273004, 42.09920106253839, 61.99663027669454, 84.92906736250268, 111.0615860420258, 140.5250750527875, 173.209.73349424783467, 249.46844055885668, 292.4867067371223, 338.5643778585117, 387.36034029093076]

Среднее значение y: 53.11374594642971

Рис. 23: 3.10(способ 1)

3.10.(способ 2):

3.10 вектор значений $y = e^x \cos(x)$ в точках $x = 3, 3.1, 3.2, \dots, 6$, найдите среднее значение y ;

```
using Statistics

# Создание вектора x и вычисление значений функции y
x_values = 3:0.1:6
y_values = map(x -> exp(x) * cos(x), x_values)

# Нахождение среднего значения y без использования импорта
mean_y = sum(y_values) / length(y_values)

# Вывод результатов
println("Значения y: ", y_values)
println("Среднее значение y: ", mean_y)
```

Значения y: [-19.884530844146987, -22.178753389342127, -24.490696732801293, -26.77318244299338, -28.969237768093574, -31.011186439374516, -32.819774760338504, -34.335.35719361853035, -35.86283371230767, -35.68773248011913, -34.68504225166807, -32.693695428321746, -29.538816297262983, -25.032529229039966, -18.975233154958957, -78, -1.3620985182057503, 10.632038010191998, 25.046704998273004, 42.09920106253839, 61.99663027669454, 84.92906736250268, 111.0615860420258, 140.5250750527875, 173.209.73349424783467, 249.46844055885668, 292.4867067371223, 338.5643778585117, 387.36034029093076]

Среднее значение y: 53.11374594642971

Рис. 24: 3.10(способ 2)

3.11.(способ 1):

3.11 вектор вида (x^i, y^j) , $x = 0.1, i = 3, 6, 9, \dots, 36, y = 0.2, j = 1, 4, 7, \dots, 34$;

```
# Создание вектора x и y
```

 $\mu = 0.1$ $y = 0.2$

```
vec1 = [(x^i, y^j) for i in 3:3:36, j in 1:3:34]
```

```
println(vec1)
```

```

0.00100000000000000002, 0.2) (0.00100000000000000002, 0.00160000000000000003) (0.00100000000000000002, 1.200000000000000005e-5) (0.00100000000000000002, 1.02400000000000000002, 0.192000000000000005e-10) (0.00100000000000000002, 0.553600000000000005e-12) (0.00100000000000000002, 5.2428000000000005e-14) (0.00100000000000000002, 4.1943400000000005e-16) (1.00000000000000000002, 3.35543200000000044e-18) (0.00100000000000000002, 2.6843545600000004e-20) (0.00100000000000000002, 2.1474836480000004e-22) (0.00100000000000000002, 0.0035e-24); (1.000000000000000004e-6, 0.2) (1.000000000000000004e-6, 0.00160000000000000003) (1.000000000000000004e-6, 1.200000000000000005e-5) (1.000000000000000004e-6, 1.02400000000000000002, 0.192000000000000005e-10) (1.000000000000000004e-6, 0.553600000000000005e-12) (1.000000000000000004e-6, 5.2428000000000005e-14) (1.000000000000000004e-6, 0.0035e-16) (1.000000000000000004e-6, 1.000000000000000004e-6, 0.192000000000000005e-10) (1.000000000000000004e-6, 0.553600000000000005e-12) (1.000000000000000004e-6, 5.2428000000000005e-14) (1.000000000000000004e-6, 0.0035e-16) (1.000000000000000004e-6, 3.35543200000000044e-18) (1.000000000000000004e-6, 2.6843545600000004e-20) (1.000000000000000004e-6, 2.1474836480000004e-22) (1.000000000000000004e-6, 0.0035e-24); (1.000000000000000005e-9, 0.2) (1.000000000000000005e-9, 0.00160000000000000003) (1.000000000000000005e-9, 1.200000000000000005e-5) (1.000000000000000005e-9, 1.02400000000000000002, 0.192000000000000005e-10) (1.000000000000000005e-9, 0.553600000000000005e-12) (1.000000000000000005e-9, 5.2428000000000005e-14) (1.000000000000000005e-9, 0.0035e-16) (1.000000000000000005e-9, 3.35543200000000044e-18) (1.000000000000000005e-9, 2.6843545600000004e-20) (1.000000000000000005e-9, 2.1474836480000004e-22) (1.000000000000000005e-9, 0.0035e-24); (1.000000000000000008e-12, 0.2) (1.000000000000000008e-12, 0.00160000000000000003) (1.000000000000000008e-12, 1.200000000000000005e-5) (1.000000000000000008e-12, 1.02400000000000000002, 0.192000000000000005e-10) (1.000000000000000008e-12, 0.553600000000000005e-12) (1.000000000000000008e-12, 5.2428000000000005e-14) (1.000000000000000008e-12, 0.0035e-16) (1.000000000000000008e-12, 3.35543200000000044e-18) (1.000000000000000008e-12, 2.6843545600000004e-20) (1.000000000000000008e-12, 2.1474836480000004e-22) (1.000000000000000008e-12, 0.0035e-24); (1.000000000000000008e-12, 1.71798691840000035e-24); (1.000000000000000008e-15, 0.2) (1.000000000000000008e-15, 0.00160000000000000003) (1.000000000000000008e-15, 1.200000000000000005e-5) (1.000000000000000008e-15, 1.02400000000000000002, 0.192000000000000005e-10) (1.000000000000000008e-15, 0.553600000000000005e-12) (1.000000000000000008e-15, 5.2428000000000005e-14) (1.000000000000000008e-15, 0.0035e-16) (1.000000000000000008e-15, 3.35543200000000044e-18) (1.000000000000000008e-15, 2.6843545600000004e-20) (1.000000000000000008e-15, 2.1474836480000004e-22) (1.000000000000000008e-15, 0.0035e-24); (1.000000000000000008e-15, 1.71798691840000035e-24); (1.000000000000000008e-18, 0.2) (1.000000000000000008e-18, 0.00160000000000000003) (1.000000000000000008e-18, 1.200000000000000005e-5) (1.000000000000000008e-18, 1.02400000000000000002, 0.192000000000000005e-10) (1.000000000000000008e-18, 0.553600000000000005e-12) (1.000000000000000008e-18, 5.2428000000000005e-14) (1.000000000000000008e-18, 0.0035e-16) (1.000000000000000008e-18, 3.35543200000000044e-18) (1.000000000000000008e-18, 2.6843545600000004e-20) (1.000000000000000008e-18, 2.1474836480000004e-22) (1.000000000000000008e-18, 0.0035e-24); (1.000000000000000008e-18, 1.71798691840000035e-24); (1.000000000000000008e-21, 0.2) (1.000000000000000008e-21, 0.00160000000000000003) (1.000000000000000008e-21, 1.200000000000000005e-5) (1.000000000000000008e-21, 1.02400000000000000002, 0.192000000000000005e-10) (1.000000000000000008e-21, 0.553600000000000005e-12) (1.000000000000000008e-21, 5.2428000000000005e-14) (1.000000000000000008e-21, 0.0035e-16) (1.000000000000000008e-21, 3.35543200000000044e-18) (1.000000000000000008e-21, 2.6843545600000004e-20) (1.000000000000000008e-21, 2.1474836480000004e-22) (1.000000000000000008e-21, 0.0035e-24); (
```


3.12.(способ 1):

3.12 вектор с элементами $\frac{2^i}{i}, i = 1, 2, \dots, M, M = 25$;

```
vec1 = [2**i / i for i in 1:25]
```

```
# Вывод результатов  
println(vec1)
```

```
[2.0, 2.0, 2.6666666666666665, 4.0, 6.4, 10.666666666666666, 18.285714285714285, 32.0, 56.888888888888886, 102.4, 186.1818181818182, 341.3333333333333, 630.1538461538461, 1428.57142, 2184.5333333333333, 4096.0, 7710.117647058823, 14563.555555555555, 27594.105263157893, 52428.8, 99864.38095238095, 190650.18181818182, 364722.0869565217, 66, 1.34217728e6]
```

Рис. 27: 3.12(способ 1)

3.12.(способ 2):

3.12 вектор с элементами $\frac{2^i}{i}, i = 1, 2, \dots, M, M = 25;$

```
vec2 = Float64[]
for i in 1:25
    push!(vec2, 2^i / i)
end
```

```
# вывод результатов
println(vec2)
```

[2.0, 2.0, 2.6666666666666665, 4.0, 6.4, 10.666666666666666, 18.285714285714285, 32.0, 56.888888888888886, 102.4, 186.1818181818182, 341.3333333333333, 630.1538461538462, 1248571428, 2184.5333333333333, 4096.0, 7710.117647058823, 14563.555555555555, 27594.105263157893, 52428.8, 99864.38095238095, 190650.18181818182, 364722.0809565217, 666.13421772806]

Рис. 28: 3.12(способ 2)

3.13.(способ 1):

3.13 вектор вида ("fn1", "fn2", .., "fnN"), $N = 30$;

```
# Первый способ создания вектора  
vec1 = ["fn%i" for i in 1:30]
```

```
# Вывод результатов  
println("Первый способ:")  
println(vec1)
```

Первый способ:

```
["fn1", "fn2", "fn3", "fn4", "fn5", "fn6", "fn7", "fn8", "fn9", "fn10", "fn11", "fn12", "fn13", "fn14", "fn15", "fn16", "fn17", "fn18", "fn19", "fn20", "fn21", "fn22", "fn23", "fn24", "fn25", "fn26", "fn27", "fn28", "fn29", "fn30"]
```

Рис. 29: 3.13(способ 1)

3.13.(способ 2):

3.13 вектор вида (“fn1”, “fn2”, ..., “fnN”), $N = 30$;

```
# Второй способ создания вектора
vec2 = String[]
for i in 1:30
    push!(vec2, "fn$i")
end
```

```
# Вывод результатов
println("\nВторой способ:")
println(vec2)
```

Второй способ:

```
["fn1", "fn2", "fn3", "fn4", "fn5", "fn6", "fn7", "fn8", "fn9", "fn10", "fn11", "fn12", "fn13", "fn14", "fn15", "fn16", "fn17", "fn18", "fn19", "fn20", "fn21", "fn22", "fn23", "fn24", "fn25", "fn26", "fn27", "fn28", "fn29", "fn30"]
```

Рис. 30: 3.13(способ 2)

3.14.1(способ 1):

3.14 векторы $x = x_1, x_2, \dots, x_n$ и $y = y_1, y_2, \dots, y_n$ целочисленного типа длины $n = 250$ как случайные выборки из совокупности $0, 1, \dots, 999$; на его основе:

3.14.1 сформируйте вектор $(y_2 - x_1, \dots, y_n - x_{n-1})$;

```
using Random

# Установим seed для воспроизводимости результатов
Random.seed!(42)

# Создание векторов x и y случайным образом
n = 250
x = rand{0:999, n}
y = rand{0:999, n}

# Первый способ формирования вектора разностей
diff_vec1 = [y[i+1] - x[i] for i in 1:(n-1)]

# Вывод результатов
println("Вектор x:")
println(x)

println("\nВектор y:")
println(y)

println("\nПервый способ вектора разностей:")
println(diff_vec1)
```

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 608, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 130, 421, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 791, 73, 695, 180, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 566, 966, 896, 480, 473, 839, 882, 643, 392, 356, 603, 330, 369, 330, 360, 608, 386, 615, 161, 670, 320, 623, 706, 476, 568, 136, 933, 148, 963, 166, 875, 638, 308]

3.14.1(способ 2):

3.14.1 сформируйте вектор $(y_2 - x_1, \dots, y_n - x_{n-1})$;

```
using Random

# Установим seed для воспроизводимости результатов
Random.seed!(42)

# Создание векторов x и y случайным образом
n = 250
x = rand{0:999, n}
y = rand{0:999, n}

# Второй способ формирования вектора разностей
diff_vec2 = Int[]
for i in 1:(n-1)
    push!(diff_vec2, y[i+1] - x[i])
end

# Вывод результатов
println("Вектор x:")
println(x)

println("\nВектор y:")
println(y)

println("\nВторой способ вектора разностей:")
println(diff_vec2)
```

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 386, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 673, 821, 68, 582, 511, 346, 998, 211, 173, 545, 100, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 47, 9, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 388, 842, 12, 341]

Вектор y:

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 544, 87, 508, 6, 1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 902, 571, 77, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 75]

3.14.2(способ 1):

3.14.2 сформируйте вектор

$$(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{n-2} + 2x_{n-1} - x_n);$$

```
using Random

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

result_vec1 = [x[i] + 2x[i+1] - x[i+2] for i in 1:(n-2)]

println("Вектор x:")
println(x)

println("\nВектор y:")
println(y)

println("\nПервый способ вектора:")
println(result_vec1)
```

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 209, 255, 878, 318, 624, 642, 480, 184, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 47, 9, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 160, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y:

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 6, 1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 982, 571, 77, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 239, 889, 815, 9, 936, 287, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 239, 889, 815, 70, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 713, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 408]

3.14.2(способ 2):

3.14.2 сформируйте вектор

$$(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{n-2} + 2x_{n-1} - x_n);$$

```
using Random

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

result_vec2 = Int[]
for i in 1:(n-2)
    push!(result_vec2, x[i] + 2x[i+1] - x[i+2])
end

println("Вектор x:")
println(x)

println("\nВектор y:")
println(y)

println("\nВторой способ вектора:")
println(result_vec2)
```

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 782, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 200, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 198, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736, 681, 867, 644, 748, 149, 227, 504, 520, 143, 4, 484, 331, 723, 118, 205, 987, 449, 758, 786, 705, 858, 538, 783, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 47, 9, 772, 450, 582, 84, 117, 900, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y:

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 809, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 540, 137, 408, 6, 1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 885, 571, 77, 496, 712, 375, 255, 107, 984, 240, 414, 185, 447, 292, 717, 165, 918, 821, 450, 663, 45, 761, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 731, 308, 308, 820, 950, 859, 295, 51, 239, 547, 414, 282, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915,

3.14.3(способ 1):

3.14.3 сформируйте вектор $\left(\frac{\sin(y_1)}{\cos(x_2)}, \frac{\sin(y_2)}{\cos(x_3)}, \dots, \frac{\sin(y_{n-1})}{\cos(x_n)} \right)$;

```
using Random
using Printf

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

result_vec1 = [sin(y[i]) / cos(x[i+1]) for i in 1:(n-1)]

@printf("Вектор x: %s\n", x)
@printf("Вектор y: %s\n", y)
@printf("Первый способ вектора: %s\n", result_vec1)
```

Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 1, 0, 79, 663, 402, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 685, 699, 353, 66, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 940, 260, 255, 878, 318, 624, 642, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 350, 560, 309, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 0, 81, 244, 673, 821, 68, 582, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 0, 31, 857, 681, 867, 644, 748, 548, 149, 227, 504, 520, 143, 4, 484, 331, 723, 118, 285, 907, 449, 758, 786, 785, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 716, 729, 772, 439, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 989, 600, 0, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 968, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 41, 6, 481, 411, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 385, 492, 222, 961, 869, 965, 214, 855, 450, 977, 500, 710, 611, 151, 0, 144, 9, 88, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 247, 796, 731, 300, 380, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 0, 39, 590, 59, 936, 267, 317, 625, 630, 524, 585, 811, 773, 458, 700, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 1, 13, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 185, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 207, 906, 592, 482, 185, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 582, 85, 468, 887, 609, 394, 408]

Первый способ вектора: [1.2867719841616385, -1.1454738309664125, -0.12889840823445167, -1.2829533770777193, -12.96493437853732, 0.9531983276634517, -0.110644918909954, 0.9355365, 1.170118873999959, -3.877544486842505, 1.3858596584019045, 0.8275930720435480, -0.6023526590132967, 1.0047180993933188, -1.2140204283000932, 3.123017888591, 152427613, -2.270052546500484, -0.47670168378140815, 3.3653169659085846, 0.05440921729995269, 2.909684001410264, 0.13242130372770972, -1.4875498748865863, -1.2963431, 807983093350086, 0.05634755616583091, -5.63371575832417, 0.5799485547485174, 0.5419216442768281, -0.5067755038924624, -69.83022305211773, -0.2894538138779738, -5.575, 0.12814562710196137, 0.33257075842588635, -1.5605474772279417, -0.5344264143708287, 0.813278295672189, -0.9500871301529229, 0.46801911628083129, 13.046072481275216, 0.535, 0.6862080248863207, 0.8170808818064438, 0.11471988820984633, 0.5870861228060275, 0.12140657015284212, 0.6805875505790745, -0.7556774221733503, -0.734397729565438, 4555626, 0.26389218012952317, -214.4222805295697, -1.72910289722121266, -0.41398555027977709, 1.1926335923910525, -1.0076823656750315, -1.00697774533484795, 0.548294261, 0.71923904447, 0.03075125905617745, -2.41271744812238153, 0.00014214863600449913, 228.8942217150278, 0.4857975107778184, -0.0, -0.6098347802224115, 3.0000000000000004, 453241, 0.93294444975184125, -1.583526528930222, 1.1510359298218504, -0.6981926076520347, 0.4780418681272226, 1.85445023340073, -1.6025211350996351, -0.0992810423381, 90544085, -1.121832198340996, -1.049554074743916, 0.8512030095777967, 0.36394351127992136, -3.0207514891207747, -0.18843087346263548, 1.0051488270679492, -2.60047254, 254544222, -2.407631324782686, 0.0008441314109087, 0.25789861625959681, 1.0001672373801179, -2.627605741675436, 0.05457370268825186, 2.455341559965681, -1.446422981

3.14.3(способ 2):

3.14.3 сформируйте вектор $\left(\frac{\sin(y_1)}{\cos(x_2)}, \frac{\sin(y_2)}{\cos(x_3)}, \dots, \frac{\sin(y_{n-1})}{\cos(x_n)} \right)$;

```
using Random
using Printf
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
result_vec2 = Float64[]
```

```
for i in 1:(n-1)
    push!(result_vec2, sin(y[i]) / cos(x[i+1]))
end
```

```
@printf("Вектор x: %s\n", x)
```

```
@printf("Вектор y: %s\n", y)
```

```
@printf("Второй способ вектора: %s\n", result_vec2)
```

Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 96, 779, 713, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 1, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 268, 318, 359, 560, 306, 915, 241, 679, 350, 943, 705, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 81, 244, 673, 821, 68, 582, 511, 346, 998, 211, 173, 545, 198, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 960, 667, 93, 799, 942, 721, 819, 285, 945, 639, 1, 31, 857, 681, 867, 644, 748, 149, 227, 504, 520, 143, 4, 484, 331, 723, 118, 265, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 4, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 582, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 0, 481, 411, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 902, 571, 88, 975, 277, 406, 712, 375, 255, 187, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 7, 247, 796, 731, 300, 380, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 913, 39, 598, 59, 936, 287, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 263, 438, 13, 492, 578, 870, 423, 951, 183, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 185, 779, 518, 392, 257, 479, 59, 346, 193, 961, 61, 907, 119, 9, 169, 764, 29, 207, 906, 592, 482, 185, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 582, 85, 468, 887, 609, 394, 408]

Первый способ вектора: [1.2867719843163035, -1.1454738309664125, -0.12889840823445167, -1.2829533770777193, -12.96493437853732, 0.9531983276634517, -0.11044491890954632, -5.09355365, 1.1701188763999999, -3.287544486842505, 1.3858596584019045, 0.8275930728435488, -0.6023526590132967, 1.0047186933933188, -1.2140204283006932, 3.123017888591526, -0.125427613, -2.270052546500484, -0.47670168378148815, 3.365311696509985846, 0.05440921279995269, 2.909684081410264, 0.13242130372778972, -1.4875496748865863, -1.29634311762803, 807983893350686, 0.05634755616583091, -5.63371575832417, 0.5799485547485174, 0.5419216442768281, -0.5967755038924624, -69.83822385211773, -0.2894538138779738, -5.575562803, 0.12814562718196137, 0.332570751842588635, -1.5605474772279417, -0.5344264143708287, 0.81337829295672189, -0.9508071301529229, 0.4680191162803129, 13.046072481275216, -0.84581535, 0.6862080240863267, 0.817080881804438, 0.11471988820984633, 0.5870861228060275, 0.21240657015284212, 0.6805875505790745, -0.7556774221733503, -0.734307727654383, -1.54556026, 0.26389026802952317, -214.4222005295697, -1.7291828972212166, -0.4139855502977789, 1.19261335923910525, -1.0070823656750315, -1.0069777453848765, 0.57801379, 0.71923904443, 0.03875125905617745, -2.4127144812238153, 0.0004124863660449913, 228.89242117150278, 0.48579775107778184, -0.0, -0.6898347882224115, 3.4914422053273855, -1.453241, 0.9329444875184125, -1.5835296228930222, 1.1510359298218504, -0.6981926076528347, 0.4780418681272226, 1.05445023348873, -1.6025211330996351, -0.6992818423385311, 1.90048085, -1.121033198384006, -1.8695540874741016, 0.85120380065777067, 0.36104311279692136, -1.0207514891207747, -0.18843087346261548, 1.0051688270670492, -2.600427546335535]

3.14.4(способ 1):

3.14.4 вычислите $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
# Первый способ вычисления суммы
```

```
sum_result1 = sum([exp(-x[i+1]) / (x[i] + 10) for i in 1:(n-1)])
```

```
println("Вектор x:")
```

```
println(x)
```

```
println("\nВектор y:")
```

```
println(y)
```

```
println("\nПервый способ суммы:")
```

```
println(sum_result1)
```

```
Вектор x:
```

```
[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 630, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 270, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 47, 9, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 410, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]
```

```
Вектор y:
```

```
[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 6, 1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 902, 571, 77, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 9, 936, 207, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 263, 438, 70, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 408]
```

```
Первый способ суммы:
```

```
0.00015970776348493994
```

3.14.4(способ 2):

3.14.4 вычислите $\sum_{i=1}^{n-1} \frac{e^{-x_i+1}}{x_i+10}$;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
sum_result2 = 0.0
```

```
for i in 1:(n-1)
```

```
    sum_result2 += exp(-x[i+1]) / (x[i] + 10)
```

```
end
```

```
println("Вектор x:")
```

```
println(x)
```

```
println("\nВектор y:")
```

```
println(y)
```

```
println("\nВторой способ суммы:")
```

```
println(sum_result2)
```

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 47, 9, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y:

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 6, 1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 902, 571, 77, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 9, 936, 207, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 254, 244, 438, 70, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 109, 704, 294, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 408]

3.14.5(способ 1):

3.14.5 выберите элементы вектора y , значения которых больше 600, и выведите на экран; определите индексы этих элементов ;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
selected_elements1 = filter(y -> y > 600, y)
```

```
println("Выбранные элементы (первый способ):")
```

```
println(selected_elements1)
```

```
# Определение индексов выбранных элементов
```

```
indices1 = findall(y -> y > 600, y)
```

```
println("\nИндексы выбранных элементов (первый способ):")
```

```
println(indices1)
```

```
Выбранные элементы (первый способ):
```

```
[828, 909, 922, 740, 700, 973, 899, 611, 946, 735, 627, 833, 908, 621, 609, 992, 833, 956, 864, 883, 813, 739, 961, 961, 869, 965, 855, 977, 710, 611, 902, 957, 975, 928, 821, 663, 762, 968, 755, 937, 796, 731, 829, 950, 859, 995, 988, 642, 888, 778, 664, 757, 889, 915, 753, 936, 625, 630, 811, 773, 708, 748, 973, 685, 744, 909, 870, 951, 672, 644, 981, 627, 872, 779, 961, 719, 811, 907, 764, 933, 906, 988, 793, 960, 700, 936, 982, 790, 887, 609]
```

```
Индексы выбранных элементов (первый способ):
```

```
[4, 6, 9, 10, 15, 16, 17, 18, 20, 23, 26, 29, 33, 34, 39, 40, 41, 45, 46, 48, 50, 51, 52, 56, 57, 58, 60, 62, 64, 65, 69, 71, 73, 76, 80, 86, 88, 89, 91, 93, 99, 101, 114, 115, 116, 125, 127, 128, 131, 132, 134, 136, 142, 143, 144, 148, 151, 152, 155, 156, 158, 161, 163, 167, 168, 173, 174, 176, 177, 185, 187, 189, 194, 195, 198, 210, 211, 212, 216, 218, 220, 224, 226, 229, 233, 234, 239, 242, 247, 248]
```

3.14.5(способ 2):

3.14.5 выберите элементы вектора y , значения которых больше 600, и выведите на экран; определите индексы этих элементов ;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
selected_elements2 = y[y .> 600]
```

```
println("\nВыбранные элементы (второй способ):")
```

```
println(selected_elements2)
```

```
indices2 = findall(y -> y > 600, y)
```

```
println("\nИндексы выбранных элементов (второй способ):")
```

```
println(indices2)
```

Выбранные элементы (второй способ):

[828, 909, 922, 740, 700, 973, 899, 611, 946, 735, 627, 833, 908, 621, 609, 992, 833, 956, 864, 883, 813, 739, 961, 961, 869, 965, 855, 977, 710, 611, 902, 957, 975, 928, 821, 663, 762, 968, 755, 937, 796, 731, 829, 950, 859, 995, 988, 642, 888, 778, 664, 757, 889, 915, 753, 936, 625, 630, 811, 773, 708, 748, 973, 685, 744, 909, 870, 951, 672, 644, 981, 627, 872, 779, 961, 719, 811, 907, 764, 933, 906, 988, 793, 960, 700, 936, 982, 790, 887, 609]

Индексы выбранных элементов (второй способ):

[4, 6, 9, 10, 15, 16, 17, 18, 20, 23, 26, 29, 33, 34, 39, 40, 41, 45, 46, 48, 50, 51, 52, 56, 57, 58, 60, 62, 64, 65, 69, 71, 73, 76, 80, 86, 88, 89, 91, 93, 99, 101, 114, 115, 116, 125, 127, 128, 131, 132, 134, 136, 142, 143, 144, 148, 151, 152, 155, 156, 158, 161, 163, 167, 168, 173, 174, 176, 177, 185, 187, 189, 194, 195, 197, 200, 210, 211, 212, 216, 218, 220, 224, 226, 229, 233, 234, 239, 242, 247, 248]

Рис. 40: 3.14.5(способ 2)

3.14.6(способ 1):

3.14.6 определите значения вектора x , соответствующие значениям вектора y , значения которых больше 600 (под соответствием понимается расположение на аналогичных индексных позициях);

```
using Random

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

selected_x1 = x[y .> 600]

println("Вектор x:")
println(x)

println("\nВектор y:")
println(y)

println("\nзначения вектора x, соответствующие y > 600 (первый способ):")
println(selected_x1)
```

клеток x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 61, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963]
791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 717, 587, 351, 86, 702, 558, 299, 532, 385, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184]
546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 953, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 786, 426, 548, 126, 833, 148, 962, 265, 875, 628, 396]
733, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 448, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736,
681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 285, 907, 449, 758, 786, 765, 858, 538, 783, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 47
9, 722, 459, 582, 84, 117, 999, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 633, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y :

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 6

3.14.7(способ 1):

3.14.7 сформируйте вектор

$\left(|x_1 - \bar{x}|^{\frac{1}{2}}, |x_2 - \bar{x}|^{\frac{1}{2}}, \dots, |x_n - \bar{x}|^{\frac{1}{2}}\right)$, где \bar{x} обозначает среднее значение вектора $x = (x_1, x_2, \dots, x_n)$;

```
using Random

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

# println
mean_x = mean(x)
result_vector = abs.(x .- mean_x) .^ (1/2)

# dufool
println("Вектор x:")
println(x)
println("\nВектор y:")
println(y)
println("\nСреднее значение вектора x: ", mean_x)
println("\nВектор:")
println(result_vector)
```

```
Вектор x:
[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 190, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 372, 30, 79, 66, 3, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 104, 963, 359, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 400, 184, 798, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 639, 736, 110, 131, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 477, 308, 716, 72, 9, 772, 459, 582, 84, 117, 980, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]
```

```
Вектор y:
[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 621, 286, 481, 41, 1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 809, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 902, 571, 957, 288, 975, 2, 77, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 306, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 809, 915, 753, 439, 590, 5, 9, 936, 287, 317, 625, 630, 524, 585, 811, 773, 458, 700, 363, 248, 748, 34, 973, 409, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 263, 438, 249, 113, 492, 3, 70, 870, 423, 951, 303, 672, 536, 403, 539, 232, 644, 981, 627, 872, 113, 185, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 188, 764, 294, 933, 207, 906, 592, 482, 105, 988, 377, 793, 127, 9, 900, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 582, 85, 468, 887, 609, 394, 408]
```

3.14.7(способ 2):

3.14.7 сформируйте вектор

$\left(|x_1 - \bar{x}|^{\frac{1}{2}}, |x_2 - \bar{x}|^{\frac{1}{2}}, \dots, |x_n - \bar{x}|^{\frac{1}{2}}\right)$, где \bar{x} обозначает среднее значение вектора $x = (x_1, x_2, \dots, x_n)$;

```
using Random
using Statistics

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

mean_x = mean(x)

result_vector = zeros(Float64, n)
for i in 1:n
    result_vector[i] = abs(x[i] - mean_x)^(1/2)
end

# Вывод результатов
println("Вектор x:")
println(x)
println("Вектор y:")
println(y)
println("\nСреднее значение вектора x: ", mean_x)
println("Вектор result_vector:")
println(result_vector)
```

```
Вектор x:
[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 372, 30, 79, 66,
3, 492, 713, 165, 527, 462, 851, 173, 296, 999, 350, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 359, 779, 711,
791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 798, 244, 44,
546, 844, 895, 689, 675, 820, 884, 643, 287, 356, 593, 338, 268, 318, 359, 568, 306, 915, 241, 679, 358, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 181, 244,
673, 821, 68, 502, 511, 346, 998, 211, 373, 545, 198, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 657, 93, 799, 842, 721, 819, 285, 945, 639, 736, 116, 131, 857,
682, 867, 684, 748, 249, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 785, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 426, 237, 477, 598, 716, 72,
9, 772, 459, 582, 84, 117, 900, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 558, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]
```

```
Вектор y:
[551, 413, 113, 828, 266, 909, 600, 8, 922, 748, 35, 337, 524, 140, 708, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 908, 621, 286, 481, 41,
1, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 901, 809, 905, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 902, 571, 957, 288, 975, 2,
77, 496, 712, 375, 255, 107, 904, 249, 414, 189, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 411, 87, 347, 128, 374, 364, 130, 755, 937, 247, 790,
731, 300, 308, 829, 956, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 402, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 590, 5,
9, 936, 287, 317, 625, 638, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 409, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 438, 249, 113, 482, 3,
798, 870, 423, 951, 189, 672, 536, 489, 139, 232, 644, 981, 627, 872, 113, 185, 778, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 987, 119, 9, 189, 704, 294, 935, 287, 906,
592, 482, 185, 988, 377, 793, 127, 9, 908, 263, 281, 423, 708, 936, 351, 282, 376, 267, 982, 234, 547, 708, 13, 582, 85, 468, 887, 609, 394, 488]
```

```
Среднее значение вектора x: 495.472
```


3.14.8(способ 1):

3.14.8 определите, сколько элементов вектора y отстоят от максимального значения не более, чем на 120 ;

```
using Random
using Statistics

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

#проверка
max_value = maximum(y)
count_close_to_max = count(elem -> abs(elem - max_value) <= 200, y)
#вывод
println("Вектор y: ", y)
println("Количество элементов вектора y, отстоящих от максимального значения не более, чем на 200: $count_close_to_max")
```

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 46, 481, 411, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 88, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 39, 590, 59, 936, 207, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 13, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 207, 906, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 408]

Количество элементов вектора y, отстоящих от максимального значения не более, чем на 200: 56

Рис. 45: 3.14.8(способ 1)

3.14.8(способ 2):

3.14.8 определите, сколько элементов вектора y отстоят от максимального значения не более, чем на 120 ;

```
using Random

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

max_value = maximum(y)
count_close_to_max = 0

for elem in y
    if abs(elem - max_value) <= 200
        count_close_to_max += 1
    end
end

println("Вектор y: ", y)
println("Количество элементов вектора y, отстоящих от максимального значения не более, чем на 200: $count_close_to_max")
```

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 4, 6, 481, 411, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 88, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 39, 590, 59, 936, 207, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 13, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 207, 906, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 408]

Количество элементов вектора y, отстоящих от максимального значения не более, чем на 200: 56

Рис. 46: 3.14.8(способ 2)

3.14.9(способ 1):

3.14.9 определите, сколько чётных и нечётных элементов вектора x ;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
even_count = count(x -> x % 2 == 0, x)
```

```
odd_count = count(x -> x % 2 != 0, x)
```

```
println("Вектор x: ", x)
```

```
println("Четные элементы в x: $even_count")
```

```
println("Нечетные элементы в x: $odd_count")
```

```
Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 81, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 31, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]
```

```
Четные элементы в x: 112
```

```
Нечетные элементы в x: 138
```

Рис. 47: 3.14.9(способ 1)

3.14.9(способ 2):

3.14.9 определите, сколько чётных и нечётных элементов вектора x ;

```
using Random

Random.seed!(42)

n = 250
x = rand(0:999, n)
y = rand(0:999, n)

even_elements = Int[]
odd_elements = Int[]

for elem in x
    if elem % 2 == 0
        push!(even_elements, elem)
    else
        push!(odd_elements, elem)
    end
end

even_count = length(even_elements)
odd_count = length(odd_elements)

println("Вектор x: ", x)
println("Четные элементы в x: $even_count")
println("Нечетные элементы в x: $odd_count")
```

```
Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321,
0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66,
779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642
244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875,
81, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945,
31, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416
716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]
Четные элементы в x: 112
Нечетные элементы в x: 138
```

3.14.10(способ 1):

3.14.10 определите, сколько элементов вектора x кратны 7 ;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
count_multiple_of_seven = sum(x .% 7 .== 0)
```

```
println("Вектор x: ", x)
```

```
println("Количество элементов вектора x, кратных 7: $count_multiple_of_seven")
```

Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 81, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 31, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 780, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Количество элементов вектора x, кратных 7: 37

Рис. 49: 3.14.10(способ 1)

3.14.10(способ 2):

3.14.10 определите, сколько элементов вектора x кратны 7 ;

```
using Random
```

```
Random.seed!(42)
```

```
n = 250
```

```
x = rand(0:999, n)
```

```
y = rand(0:999, n)
```

```
count_multiple_of_seven = count(x -> x % 7 == 0, x)
```

```
println("Вектор x: ", x)
```

```
println("Количество элементов вектора x, кратных 7: $count_multiple_of_seven")
```

Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 81, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 31, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 750, 786, 705, 850, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Количество элементов вектора x, кратных 7: 37

Рис. 50: 3.14.10(способ 2)

3.14.11(способ 1):

3.14.11 отсортируйте элементы вектора x в порядке возрастания элементов вектора y ;

```
using Random

Random.seed!(42)

n = 250

# Создание векторов x и y случайных целых чисел от 0 до 999
x = rand(0:999, n)
y = rand(0:999, n)

# Способ 1: Сортировка с использованием zip
sorted_indices_2 = sort(1:n, by = i -> y[i])
sorted_x1 = x[sorted_indices_2]

# Вывод результатов
println("Вектор x: ", x)
println("Вектор y: ", y)
println("Отсортированный вектор x по возрастанию элементов вектора y (способ 1): ", sorted_x1)
```

Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 81, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 31, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 4, 6, 481, 411, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 88, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 39, 590, 59, 936, 207, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 13, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 207, 906, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 930, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 53, 48, 397, 477, 205, 131, 567, 663, 318, 480, 80, 265, 143, 267, 66, 515, 699, 966, 351, 416, 148, 739, 741, 338, 716, 68, 93, 954, 344, 4, 216, 4, 287, 448, 244, 18, 623, 705, 639, 945, 202, 673, 550, 962, 791, 875, 429, 782, 757, 560, 30, 779, 771, 477, 884, 844, 548, 895, 502, 173, 639, 703, 878, 561, 945, 642, 681, 462, 624,

3.14.11(способ 2):

3.14.11 отсортируйте элементы вектора x в порядке возрастания элементов вектора y ;

```
using Random

Random.seed!(42)

n = 250

x = rand(0:999, n)
y = rand(0:999, n)

sorted_indices = sortperm(y)
sorted_x2 = x[sorted_indices]

println("Вектор x: ", x)
println("Вектор y: ", y)
println("Отсортированный вектор x по возрастанию элементов вектора y (способ 2): ", sorted_x2)
```

Вектор x: [629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 0, 79, 663, 492, 713, 165, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 619, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 779, 711, 791, 73, 695, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 299, 532, 835, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 244, 44, 546, 844, 895, 689, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 81, 244, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 48, 439, 227, 484, 353, 966, 667, 93, 799, 942, 721, 819, 285, 945, 31, 857, 681, 867, 644, 748, 149, 227, 594, 520, 143, 4, 484, 331, 723, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 4, 6, 481, 411, 34, 609, 992, 833, 373, 65, 321, 956, 864, 251, 883, 282, 813, 739, 961, 305, 492, 222, 961, 869, 965, 214, 855, 450, 977, 508, 710, 611, 151, 0, 144, 88, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 234, 393, 968, 451, 87, 347, 128, 374, 364, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 478, 995, 283, 988, 642, 435, 545, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 4, 39, 590, 59, 936, 207, 317, 625, 630, 524, 585, 811, 773, 458, 708, 363, 248, 748, 34, 973, 499, 5, 499, 685, 744, 415, 162, 13, 211, 909, 644, 522, 887, 673, 263, 13, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 207, 906, 592, 482, 105, 988, 377, 793, 127, 9, 960, 263, 281, 423, 700, 936, 351, 282, 376, 267, 982, 234, 547, 790, 13, 502, 85, 468, 887, 609, 394, 408]

Отсортированный вектор x по возрастанию элементов вектора y (способ 2): [353, 48, 668, 384, 485, 667, 216, 492, 46, 661, 532, 643, 673, 538, 31, 853, 70, 255, 116, 5, 48, 397, 477, 131, 205, 567, 663, 318, 480, 80, 265, 143, 267, 66, 515, 699, 966, 351, 416, 148, 739, 741, 338, 68, 716, 93, 954, 344, 4, 4, 216, 287, 448, 244, 18, 623, 705, 945, 639, 202, 673, 550, 791, 962, 875, 429, 757, 782, 560, 30, 779, 771, 477, 884, 844, 548, 895, 502, 173, 639, 703, 878, 561, 945, 642, 681, 462, 624, 6, 793, 12, 260, 520, 341, 663, 450, 937, 593, 353, 644, 776, 833, 241, 157, 736, 381, 415, 654, 558, 269, 190, 93, 318, 858, 79, 459, 619, 706, 857, 954, 428, 439, 58, 600, 721, 342, 998, 594, 679, 321, 356, 623, 629, 423, 963, 434, 463, 858, 211, 563, 244, 772, 613, 713, 842, 191, 686, 372, 511, 315, 723, 346, 998, 744, 184, 1, 85, 227, 515, 28, 172, 847, 695, 587, 399, 546, 196, 279, 299, 484, 466, 746, 184, 126, 835, 237, 545, 943, 449, 769, 909, 44, 173, 858, 619, 702, 703, 689, 319, 527, 9, 81, 867, 118, 731, 819, 308, 350, 628, 272, 194, 435, 729, 915, 721, 165, 799, 398, 457, 86, 398, 821, 419, 798, 484, 673, 748, 296, 359, 836, 320, 111, 779, 835]

3.14.12(способ 1):

3.14.12 выведите элементы вектора x , которые входят в десятку наибольших (top-10)?

```
using Random

Random.seed!(42)

n = 250

x = rand(0:999, n)
y = rand(0:999, n)

top_10_indices = sortperm(x, rev=true)[1:10]
top_10_elements = x[top_10_indices]

println("Top 10 элементов вектора x:")
println(top_10_elements)

Top 10 элементов вектора x:
[999, 998, 973, 966, 963, 962, 954, 949, 948, 945]
```

Рис. 53: 3.14.12(способ 1)

3.14.12(способ 2):

3.14.12 выведите элементы вектора x , которые входят в десятку наибольших (top-10)?

```
using Random

Random.seed!(42)

n = 250

x = rand(0:999, n)
y = rand(0:999, n)

top_10_elements = sort(x)[end:-1:end-9]

println("Top 10 элементов вектора x:")
println(top_10_elements)

Top 10 элементов вектора x:
[999, 998, 973, 966, 963, 962, 954, 949, 948, 945]
```

Рис. 54: 3.14.12(способ 2)

3.14.13(способ 1):

3.14.13 сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x

```
using Random

Random.seed!(42)

n = 250

x = rand(0:999, n)
y = rand(0:999, n)

unique_x = unique(x)

println("Вектор x с уникальными элементами:")
println(unique_x)
```

Вектор x с уникальными элементами:
[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 372, 2, 713, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 359, 779, 711, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 532, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 798, 244, 44, 546, 844, 884, 643, 287, 593, 338, 260, 560, 306, 915, 241, 679, 350, 943, 706, 426, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 821, 68, 502, 511, 346, 998, 211, 545, 8, 466, 46, 483, 48, 439, 227, 966, 667, 93, 799, 942, 819, 285, 736, 110, 131, 857, 681, 867, 644, 748, 149, 594, 520, 143, 331, 723, 118, 205, 907, 449, 758, 786, 31, 399, 567, 384, 416, 237, 716, 729, 772, 459, 582, 84, 117, 909, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 769, 701, 70, 308, 842, 12, 341]

Рис. 55: 3.14.13(способ 1)

сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x

3.14.13(способ 2):

3.14.13 сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x

```
using Random

Random.seed!(42)

n = 250

x = rand(0:999, n)
y = rand(0:999, n)

unique_x = []
for element in x
    if !(element in unique_x)
        push!(unique_x, element)
    end
end

println("Вектор x с уникальными элементами:")
println(unique_x)
```

Вектор x с уникальными элементами:

Any[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 272, 191, 423, 484, 80, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 37, 492, 713, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 279, 320, 548, 344, 111, 81, 835, 954, 218, 654, 281, 364, 847, 686, 699, 353, 66, 194, 963, 359, 779, 7, 5, 100, 623, 948, 435, 973, 937, 739, 415, 771, 587, 351, 86, 702, 558, 532, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 624, 642, 480, 184, 798, 244, 44, 546, 1, 20, 884, 643, 287, 593, 338, 260, 560, 306, 915, 241, 679, 350, 943, 706, 426, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 821, 68, 502, 511, 346, 998, 211, 5, 5, 18, 466, 46, 483, 48, 439, 227, 966, 667, 93, 799, 942, 819, 285, 736, 118, 131, 857, 681, 867, 644, 748, 149, 594, 520, 143, 331, 723, 118, 205, 907, 449, 758, 1, 41, 31, 399, 567, 384, 416, 237, 716, 729, 772, 459, 582, 84, 117, 909, 485, 836, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 769, 701, 70, 308, 842, 12, 34]

Рис. 56: 3.14.13(способ 2)

4. Создаю массив squares, в котором будут храниться квадраты всех целых чисел от 1 до 120.

```
# Создаем массив squares с квадратами целых чисел от 1 до 100  
squares = [i**2 for i in range(1,101)]
```

```
# Выводим результат  
print(squares)
```

```
[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. 57: Массив squares

5. Подключаю пакет Primes (функции для вычисления простых чисел). Сгенерирую массив `myprimes`, в котором будут храниться первые 168 простых чисел. Определяю 89-е наименьшее простое число. Получаю срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа.

```
# Подключаем пакет Primes
using Primes

# Генерируем массив myprimes с первыми 168 простыми числами
myprimes = primes(1000)[1:168]

# Определяем 89-е наименьшее простое число
diopoz = myprimes[89]

# Получаем срез массива с 89-го до 99-го элемента включительно
slice_of_primes = myprimes[89:99]

# Выводим результаты
println("Массив myprimes: ", myprimes)
println("89-е наименьшее простое число: ", diopoz)
println("Срез с 89-го до 99-го элемента: ", slice_of_primes)
```

Массив 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, 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, 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, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]

89-е наименьшее простое число: 461

Срез с 89-го до 99-го элемента: [461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523]

6. Вычисляю следующие выражения:

$$6.1 \sum_{i=10}^{120} (i^3 + 4i^2);$$

```
# Вычисляем сумму выражения  $\sum (i^3 + 4i^2)$  для  $i$  от 10 до 100
result = sum(i^3 + 4*i^2 for i in range(10,101))

# Выводим результат
println("Результат выражения: ", result)

Результат выражения: 26852735
```

Рис. 59: Формула 6.1

6.2:

$$6.2 \sum_{i=1}^M \left(\frac{2^i}{i} + \frac{3^i}{i^2} \right), M = 25;$$

```
# Установим значение M
M = 25

# Вычислим сумму выражения  $\sum (2^i/i + 3^i/i^2)$  для i от 1 до M
result = sum((2**i/i) + (3**i/i**2) for i in range(1,M+1))

# Выводим результат
println("Результат выражения: ", result)
```

Результат выражения: 2.1291704368143802e9

Рис. 60: Формула 6.2

6.3:

$$6.3 \quad 1 + \frac{2}{3} + \left(\frac{2}{3} \frac{4}{5}\right) + \left(\frac{2}{3} \frac{4}{5} \frac{6}{7}\right) + \dots + \left(\frac{2}{3} \frac{4}{5} \dots \frac{38}{39}\right).$$

```
# Установим значение N
N = 20

# Вычислим сумму выражения 1 + 2/3 + (2/3 * 4/5) + ... + (2/3 * 4/5 * ... * 38/39)
result = 1.0 + sum(prod([(2 * i)/(2 * i + 1) for i in range(1, n)]) for n in range(1, N))

# Выводим результат
println("Результат выражения: ", result)
```

Результат выражения: 7.170891165651219

Рис. 61: Формула 6.3

Выводы

Изучены несколько структур данных, реализованных в Julia, научился применять их и операции над ними для решения задач