

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Компьютерный практикум

по статистическому анализу данных

Студент: Евдокимов Иван Андреевич

Группа: НФИбд-01-20

МОСКВА

2023 г.

Цель работы:

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

In [1]: ;date

Tue Nov 14 21:15:03 RTZ 2023

In [2]: ;whoami

win-or11fa978ms\admin

1. Даны множества: $A = \{0, 3, 4, 9\}$, $B = \{1, 3, 4, 7\}$, $C = \{0, 1, 2, 4, 7, 8, 9\}$. Найти $P = A \cap B \cup A \cap B \cup A \cap C \cup B \cap C$.

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

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

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

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

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

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

Пример 1: Множество целых чисел и множество строк

```
In [4]: # 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"]

Пример 2: Множество дробных чисел и множество символов

```
In [5]: # Создаю 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'])

Пример 3: Множество булевых значений и множество кортежей

```
In [6]: # 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")])

3. Создайте разными способами:

- 3.1) массив $(1, 2, 3, \dots, N-1, N)$, N выберите больше 20 ;
- 3.2) массив $(N, N-1, \dots, 2, 1)$, N выберите больше 20 ;
- 3.3) массив $(1, 2, 3, \dots, N-1, N, N-1, \dots, 2, 1)$, N выберите больше 20 ;
- 3.4) массив с именем `tmp` вида $(4, 6, 3)$;
- 3.5) массив, в котором первый элемент массива `tmp` повторяется 10 раз;
- 3.6) массив, в котором все элементы массива `tmp` повторяются 10 раз;
- 3.7) массив, в котором первый элемент массива `tmp` встречается 11 раз, второй элемент - 10 раз, третий элемент - 10 раз;
- 3.8) массив, в котором первый элемент массива `tmp` встречается 10 раз подряд, второй элемент - 20 раз подряд, третий элемент - 30 раз подряд;
- 3.9) массив из элементов вида $2^{t \cdot m \cdot p[i]}$, $i=1, 2, 3$, где элемент $2^{t \cdot m \cdot p[3]}$ встречается 4 раза; посчитайте в полученном векторе, сколько раз встречается цифра 6, и выведите это значение на экран;
- 3.10) вектор значений $y = e^x \cos(x)$ в точках $x = 3, 3.1, 3.2, \dots, 6$, найдите среднее значение y ;
- 3.11) вектор вида (x^i, y^j) , $x=0.1$, $i=3, 6, 9, \dots, 36$, $y=0.2$, $j=1, 4, 7, \dots, 34$;
- 3.12) вектор с элементами $\frac{2^i}{i}$, $i=1, 2, \dots, M$, $M=25$;
- 3.13) вектор вида (f_1, f_2, \dots, f_N) , $N=30$;
- 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:

- сформируйте вектор $\left(y_2 - x_1, \dots, y_n - x_{n-1}\right)$; (3.14.1)
- сформируйте вектор $\left(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{n-2} + 2x_{n-1} - x_n\right)$; (3.14.2)
- сформируйте вектор $\left(\frac{\sin(y_1)}{\cos(x_2)}, \frac{\sin(y_2)}{\cos(x_3)}, \dots, \frac{\sin(y_{n-1})}{\cos(x_n)}\right)$; (3.14.3)
- вычислите $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$; (3.14.4)
- выберите элементы вектора y , значения которых больше 600, и выведите на экран; определите индексы этих элементов; (3.14.5)
- определите значения вектора x , соответствующие значениям вектора y , значения которых больше 600 (под соответствием понимается расположение на аналогичных индексных позициях); (3.14.6)
- сформируйте вектор $\left(\left|x_1 - \bar{x}\right|^{\frac{1}{2}}, \left|x_2 - \bar{x}\right|^{\frac{1}{2}}, \dots, \left|x_n - \bar{x}\right|^{\frac{1}{2}}\right)$, где \bar{x} обозначает среднее значение вектора $x = (x_1, x_2, \dots, x_n)$; (3.14.7)
- определите, сколько элементов вектора y отстоят от максимального значения не более, чем на 200; (3.14.8)
- определите, сколько чётных и нечётных элементов вектора x ; (3.14.9)
- определите, сколько элементов вектора x кратны 7; (3.14.10)
- отсортируйте элементы вектора x в порядке возрастания элементов вектора y ; (3.14.11)
- выведите элементы вектора x , которые входят в десятку наибольших (top-10)?; (3.14.12)
- сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x . (3.14.13)

3.1 (способ 1)

In [7]: `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]

3.1 (способ 2)

In [8]: `N = 25 # Замените это значение на любое число больше 20`

```
arr = 1:N # Создание массива от 1 до N
arr = collect(arr) # Преобразование в явный 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]

3.2 (способ 1)

```
In [9]: 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]
```

3.2 (способ 2)

```
In [10]: N = 25 # Замените это значение на любое число больше 20

arr = reverse([i for i in 1:N]) # Создание массива от N до 1 и затем его обращение
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]
```

3.3 (способ 1)

```
In [11]: N = 25 # Замените это значение на любое число больше 20

arr = [1:N; N-1:-1:1] # Создание массива, объединяя две последовательности
println(arr)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 2
3, 24, 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]
```

3.3 (способ 2)

```
In [12]: 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, 2
3, 24, 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]
```

3.4 (способ 1)

```
In [13]: tmp = [4, 6, 3]
println(tmp)
```

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

3.4 (способ 2)

```
In [14]: tmp = Int[]
push!(tmp, 4)
push!(tmp, 6)
push!(tmp, 3)
println(tmp)
```

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

3.5 (способ 1)

```
In [15]: tmp = [4, 6, 3]
first_element = tmp[1]
arr1 = [first_element for _ in 1:10]
println(arr1)
```

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

3.5 (способ 2)

```
In [16]: 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]

3.6 (способ 1)

```
In [17]: tmp = [4, 6, 3]
arr1_repeated = [elem for elem in tmp for _ in 1:10]
println(arr1_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]

3.6 (способ 2)

```
In [18]: tmp = [4, 6, 3]
arr2_repeated = repeat(tmp, inner=10)
println(arr2_repeated)
```

[4, 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]

3.7 (способ 1)

```
In [19]: 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 1:30]
println(tmp_repeated)
```

[4, 4, 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]

3.7 (способ 2)

```
In [20]: 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, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

3.8 (способ 1)

```
In [21]: 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 1:30]
println(tmp_repeated)
```

[illegible]

3.8 (способ 2)

```
In [22]: tmp = [4, 6, 3] # Замените это значение на ваши элементы
tmp_repeated = [fill(tmp[1], 11); fill(tmp[2], 20); fill(tmp[3], 30)]
println(tmp_repeated)
```

[illegible]

3.9 (способ 1)

```
In [23]: # 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

3.9 (способ 2)

```
In [24]: # 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

3.10 (способ 1)

In [25]: `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, -24.490696732801293, -26.77318244299338, -28.969237768093574, -31.011186439374516, -32.819774760338504, -34.30336011037369, -35.35719361853035, -35.86283371230767, -35.68773248011913, -34.68504225166807, -32.693695428321746, -29.538816297262983, -25.032529229039966, -18.975233154958957, -11.157417389647478, -1.3620985182057503, 10.632038010191998, 25.046704998273004, 42.09920106253839, 61.99663027669454, 84.92906736250268, 111.0615860420258, 140.5250750527875, 173.40577640857734, 209.73349424783467, 249.46844055885668, 292.4867067371223, 338.5643778585117, 387.36034029093076]

Среднее значение y: 53.11374594642971

3.10 (способ 2)

In [26]: `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, -34.30336011037369, -35.35719361853035, -35.86283371230767, -35.68773248011913, -34.68504225166807, -32.693695428321746, -29.538816297262983, -25.032529229039966, -18.975233154958957, -11.157417389647478, -1.3620985182057503, 10.632038010191998, 25.046704998273004, 42.09920106253839, 61.99663027669454, 84.92906736250268, 111.0615860420258, 140.5250750527875, 173.40577640857734, 209.73349424783467, 249.46844055885668, 292.4867067371223, 338.5643778585117, 387.36034029093076]

Среднее значение y: 53.11374594642971

3.11 (способ 1)

In [27]: `# Создание вектора x и y`

```

x = 0.1
y = 0.2

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

```



```
println(vec1)
```

9/49

-27, 3.35544320000000044e-18) (1.00000000000000015e-27, 2.6843545600000004e-20) (1.00000000000000015e-27, 2.1474836480000004e-22) (1.00000000000000015e-27, 1.7179869184000035e-24); (1.00000000000000017e-30, 0.2) (1.00000000000000017e-30, 0.0016000000000000003) (1.00000000000000017e-30, 1.2800000000000006e-5) (1.00000000000000017e-30, 1.0240000000000006e-7) (1.00000000000000017e-30, 8.192000000000005e-10) (1.00000000000000017e-30, 6.5536000000000055e-12) (1.00000000000000017e-30, 5.2428800000000056e-14) (1.00000000000000017e-30, 4.194304000000005e-16) (1.00000000000000017e-30, 3.35544320000000044e-18) (1.00000000000000017e-30, 2.6843545600000004e-20) (1.00000000000000017e-30, 2.1474836480000004e-22) (1.00000000000000017e-30, 1.7179869184000035e-24); (1.00000000000000018e-33, 0.2) (1.00000000000000018e-33, 0.0016000000000000003) (1.00000000000000018e-33, 1.2800000000000006e-5) (1.00000000000000018e-33, 1.0240000000000006e-7) (1.00000000000000018e-33, 8.192000000000005e-10) (1.00000000000000018e-33, 6.5536000000000055e-12) (1.00000000000000018e-33, 5.2428800000000056e-14) (1.00000000000000018e-33, 4.194304000000005e-16) (1.00000000000000018e-33, 3.35544320000000044e-18) (1.00000000000000018e-33, 2.6843545600000004e-20) (1.00000000000000018e-33, 2.1474836480000004e-22) (1.00000000000000018e-33, 1.7179869184000035e-24); (1.0000000000000002e-36, 0.2) (1.0000000000000002e-36, 0.0016000000000000003) (1.0000000000000002e-36, 1.2800000000000006e-5) (1.0000000000000002e-36, 1.0240000000000006e-7) (1.0000000000000002e-36, 8.192000000000005e-10) (1.0000000000000002e-36, 6.5536000000000055e-12) (1.0000000000000002e-36, 5.2428800000000056e-14) (1.0000000000000002e-36, 4.194304000000005e-16) (1.0000000000000002e-36, 3.35544320000000044e-18) (1.0000000000000002e-36, 2.6843545600000004e-20) (1.0000000000000002e-36, 2.1474836480000004e-22) (1.0000000000000002e-36, 1.7179869184000035e-24)]

3.11 (способ 2)

```
In [28]: x = 0.1
y = 0.2

vec2 = []
for i in 3:3:36
    for j in 1:3:34
        push!(vec2, (x^i, y^j))
    end
end

println(vec2)
```

Any[(0.0010000000000000002, 0.2), (0.0010000000000000002, 0.0016000000000000003), (0.0010000000000000002, 1.2800000000000006e-5), (0.0010000000000000002, 1.0240000000000006e-7), (0.0010000000000000002, 8.192000000000005e-10), (0.0010000000000000002, 6.5536000000000055e-12), (0.0010000000000000002, 5.2428800000000056e-14), (0.0010000000000000002, 4.194304000000005e-16), (0.0010000000000000002, 3.35544320000000044e-18), (0.0010000000000000002, 2.684354560000004e-20), (0.00100000000000000000002, 2.147483648000004e-22), (0.0010000000000000002, 1.7179869184000035e-24), (1.0000000000000004e-6, 0.2), (1.0000000000000004e-6, 0.0016000000000000003), (1.0000000000000004e-6, 1.2800000000000006e-5), (1.0000000000000004e-6, 1.0240000000000006e-7), (1.0000000000000004e-6, 8.192000000000005e-10), (1.0000000000000004e-6, 6.5536000000000055e-12), (1.0000000000000004e-6, 5.2428800000000056e-14), (1.0000000000000004e-6, 4.194304000000005e-16), (1.0000000000000004e-6, 3.35544320000000044e-18), (1.0000000000000004e-6, 2.684354560000004e-20), (1.0000000000000004e-6, 2.147483648000004e-22), (1.0000000000000004e-6, 1.7179869184000035e-24), (1.0000000000000005e-9, 0.2), (1.0000000000000005e-9, 0.0016000000000000003), (1.0000000000000005e-9, 1.2800000000000006e-5), (1.0000000000000005e-9, 1.0240000000000006e-7), (1.0000000000000005e-9, 8.192000000000005e-10), (1.0000000000000005e-9, 6.5536000000000055e-12), (1.0000000000000005e-9, 5.2428800000000056e-14), (1.0000000000000005e-9, 4.194304000000005e-16), (1.0000000000000005e-9, 3.35544320000000044e-18), (1.0000000000000005e-9, 2.684354560000004e-20), (1.0000000000000005e-9, 2.147483648000004e-22), (1.0000000000000005e-9, 1.7179869184000035e-24), (1.0000000000000008e-12, 0.2), (1.0000000000000008e-12, 0.0016000000000000003), (1.0000000000000008e-12, 1.2800000000000006e-5), (1.0000000000000008e-12, 1.0240000000000006e-7), (1.0000000000000008e-12, 8.192000000000005e-10), (1.0000000000000008e-12, 6.5536000000000055e-12), (1.0000000000000008e-12, 5.2428800000000056e-14), (1.0000000000000008e-12, 4.194304000000005e-16), (1.0000000000000008e-12, 3.35544320000000044e-18), (1.0000000000000008e-12, 2.684354560000004e-20), (1.0000000000000008e-12, 2.147483648000004e-22), (1.0000000000000008e-12, 1.7179869184000035e-24), (1.0000000000000009e-15, 0.2), (1.0000000000000009e-15, 0.0016000000000000003), (1.0000000000000009e-15, 1.2800000000000006e-5), (1.0000000000000009e-15, 1.0240000000000006e-7), (1.0000000000000009e-15, 8.192000000000005e-10), (1.0000000000000009e-15, 6.5536000000000055e-12), (1.0000000000000009e-15, 5.2428800000000056e-14), (1.0000000000000009e-15, 4.194304000000005e-16), (1.0000000000000009e-15, 3.35544320000000044e-18), (1.0000000000000009e-15, 2.684354560000004e-20), (1.0000000000000009e-15, 2.147483648000004e-22), (1.0000000000000009e-15, 1.7179869184000035e-24), (1.0000000000000008e-18, 0.2), (1.0000000000000008e-18, 0.0016000000000000003), (1.0000000000000008e-18, 1.2800000000000006e-5), (1.0000000000000008e-18, 1.0240000000000006e-7), (1.0000000000000008e-18, 8.192000000000005e-10), (1.0000000000000008e-18, 6.5536000000000055e-12), (1.0000000000000008e-18, 5.2428800000000056e-14), (1.0000000000000008e-18, 4.194304000000005e-16), (1.0000000000000008e-18, 3.35544320000000044e-18), (1.0000000000000008e-18, 2.684354560000004e-20), (1.0000000000000008e-18, 2.147483648000004e-22), (1.0000000000000008e-18, 1.7179869184000035e-24), (1.0000000000000012e-21, 0.2), (1.0000000000000012e-21, 0.0016000000000000003), (1.0000000000000012e-21, 1.2800000000000006e-5), (1.0000000000000012e-21, 1.0240000000000006e-7), (1.0000000000000012e-21, 8.192000000000005e-10), (1.0000000000000012e-21, 6.5536000000000055e-12), (1.0000000000000012e-21, 5.2428800000000056e-14), (1.0000000000000012e-21, 4.194304000000005e-16), (1.0000000000000012e-21, 3.35544320000000044e-18), (1.0000000000000012e-21, 2.684354560000004e-20), (1.0000000000000012e-21, 2.147483648000004e-22), (1.0000000000000012e-21, 1.7179869184000035e-24), (1.0000000000000012e-24, 0.2), (1.0000000000000012e-24, 0.0016000000000000003), (1.0000000000000012e-24, 1.2800000000000006e-5), (1.0000000000000012e-24, 1.0240000000000006e-7), (1.0000000000000012e-24, 8.192000000000005e-10), (1.0000000000000012e-24, 6.5536000000000055e-12), (1.0000000000000012e-24, 5.2428800000000056e-14), (1.0000000000000012e-24, 4.194304000000005e-16), (1.0000000000000012e-24, 3.35544320000000044e-18), (1.0000000000000012e-24, 2.684354560000004e-20), (1.0000000000000012e-24, 2.147483648000004e-22), (1.0000000000000012e-24, 1.7179869184000035e-24), (1.0000000000000015e-27, 0.2), (1.0000000000000015e-27, 0.0016000000000000003), (1.0000000000000015e-27, 1.2800000000000006e-5), (1.0000000000000015e-27, 1.0240000000000006e-7), (1.0000000000000015e-27, 8.192000000000005e-10), (1.0000000000000015e-27, 6.5536000000000055e-12), (1.000000000000001

5e-27, 5.24288000000000056e-14), (1.00000000000000015e-27, 4.1943040000000005e-16), (1.00000000000000015e-27, 3.35544320000000044e-18), (1.00000000000000015e-27, 2.6843545600000004e-20), (1.00000000000000015e-27, 2.1474836480000004e-22), (1.00000000000000015e-27, 1.7179869184000035e-24), (1.00000000000000017e-30, 0.2), (1.00000000000000017e-30, 0.0016000000000000003), (1.00000000000000017e-30, 1.2800000000000006e-5), (1.00000000000000017e-30, 1.0240000000000006e-7), (1.00000000000000017e-30, 8.1920000000000005e-10), (1.00000000000000017e-30, 6.55360000000000055e-12), (1.00000000000000017e-30, 5.24288000000000056e-14), (1.00000000000000017e-30, 4.1943040000000005e-16), (1.00000000000000017e-30, 3.35544320000000044e-18), (1.00000000000000017e-30, 2.6843545600000004e-20), (1.00000000000000017e-30, 2.1474836480000004e-22), (1.00000000000000017e-30, 1.7179869184000035e-24), (1.00000000000000018e-33, 0.2), (1.00000000000000018e-33, 0.0016000000000000003), (1.00000000000000018e-33, 1.2800000000000006e-5), (1.00000000000000018e-33, 1.0240000000000006e-7), (1.00000000000000018e-33, 8.1920000000000005e-10), (1.00000000000000018e-33, 6.55360000000000055e-12), (1.00000000000000018e-33, 5.24288000000000056e-14), (1.00000000000000018e-33, 4.1943040000000005e-16), (1.00000000000000018e-33, 3.35544320000000044e-18), (1.00000000000000018e-33, 2.6843545600000004e-20), (1.00000000000000018e-33, 2.1474836480000004e-22), (1.00000000000000018e-33, 1.7179869184000035e-24), (1.0000000000000002e-36, 0.2), (1.0000000000000002e-36, 0.0016000000000000003), (1.0000000000000002e-36, 1.2800000000000006e-5), (1.0000000000000002e-36, 1.0240000000000006e-7), (1.0000000000000002e-36, 8.1920000000000005e-10), (1.0000000000000002e-36, 6.55360000000000055e-12), (1.0000000000000002e-36, 5.24288000000000056e-14), (1.0000000000000002e-36, 4.1943040000000005e-16), (1.0000000000000002e-36, 3.35544320000000044e-18), (1.0000000000000002e-36, 2.6843545600000004e-20), (1.0000000000000002e-36, 2.1474836480000004e-22), (1.0000000000000002e-36, 1.7179869184000035e-24)]

3.12 (способ 1)

In [29]: `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, 630.1538461538462, 1170.2857142857142, 2184.5333333333333, 4096.0, 7710.117647058823, 14563.555555555555, 27594.105263157893, 52428.8, 99864.38095238095, 190650.18181818182, 364722.0869565217, 699050.6666666666, 1.34217728e6]

3.12 (способ 2)

```
In [30]: 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, 1170.2857142857142, 2184.5333333333333, 4096.0, 7710.117647058823, 14563.555555555555, 27594.105263157893, 52428.8, 99864.38095238095, 190650.18181818182, 364722.0869565217, 699050.6666666666, 1.34217728e6]

3.13 (способ 1)

```
In [31]: # 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"]
```

3.13 (способ 2)

```
In [32]: # 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"]
```

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

```
In [33]: 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, 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, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Первый способ вектора разностей:

[-216, -337, 351, -437, 236, 435, -605, 254, 283, -264, -324, -115, -202, 433, 458, 809, 339, 374, 523, -352, 66, 220, 323, -489, 230, 263, 148, -25, 268, -17, 116, 751, -100, -86, 451, 332, -629, 117, 279, 668, -154, -397, -532, 783, 568, -748, 527, -449, 56, 120, 682, -15, -56, -397, 617, 758, 884, -621, -99, 232, 323, 27, 346, -236, -535, -699, -209, 836, 377, -6, -71, 196, -434, -295, 639, -320, 155, -516, -44, -186, -559, -752, -292, -123, -54, -422, 577, 735, -252, 105, -254, 230, -257, -371, 124, -214, 389, 175, -498, -182, 92, -750, 56, -260, -512, 275, 753, -551, 552, 687, -246, -536, -66, 261, 186, -525, -833, -404, 260, 58, -391, 65, 210, 677, -76, 428, 336, -480, 304, 209, 428, -451, -42, 91, 209, 302, -652, 129, -827, 13, 14, 287, 355, -307, 209, -185, 263, -614, 249, 123, 119, 178, -413, 600, 600, -87, 518, 191, -697, 730, -432, 927, -280, -478, 451, 246, 517, -69, -191, -953, -456, 816, -155, -420, 166, -146, -22, -682, -201, -487, 3, 361, -487, 189, -444, 307, -645, 523, 309, -191, -381, 89, 640, 497, 296, 149, -5, -100, -128, 69, -366, -529, -226, -799, -192, -510, 220, -716, 688, 412, 49, -796, -558, -215, 348, 57, 456, -191, 190, -137, -290, -354, 406, 293, 676, -782, -654, 475, -573, 79, -6, -76, 908, -68, -279, -406, 200, 432, 65, 331, 167, -756, 286, -616, 398, 794, 301, -448, 396]

3.14 (способ 2)(3.14.1) - сформируйте вектор $\left(y_2 - x_1, \dots, y_n - x_{n-1}\right)$;

In [34]: `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, 372, 30, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Второй способ вектора разностей:

[-216, -337, 351, -437, 236, 435, -605, 254, 283, -264, -324, -115, -202, 433, 458, 809, 339, 374, 523, -352, 66, 220, 323, -489, 230, 263, 148, -25, 268, -17, 116, 751, -100, -86, 451, 332, -629, 117, 279, 668, -154, -397, -532, 783, 568, -748, 527, -449, 56, 120, 682, -15, -56, -397, 617, 758, 884, -621, -99, 232, 323, 27, 346, -236, -535, -699, -209, 836, 377, -6, -71, 196, -434, -295, 639, -320, 155, -516, -44, -186, -559, -752, -292, -123, -54, -422, 577, 735, -252, 105, -254, 230, -257, -371, 124, -214, 389, 175, -498, -182, 92, -750, 56, -260, -512, 275, 753, -551, 552, 687, -246, -536, -66, 261, 186, -525, -833, -404, 260, 58, -391, 65, 210, 677, -76, 428, 336, -480, 304, 209, 428, -451, -42, 91, 209, 302, -652, 129, -827, 13, 14, 287, 355, -307, 209, -185, 263, -614, 249, 123, 119, 178, -413, 600, 600, -87, 518, 191, -697, 730, -432, 927, -280, -478, 451, 246, 517, -69, -191, -953, -456, 816, -155, -420, 166, -146, -22, -682, -201, -487, 3, 361, -487, 189, -444, 307, -645, 523, 309, -191, -381, 89, 640, 497, 296, 149, -5, -100, -128, 69, -366, -529, -226, -799, -192, -510, 220, -716, 688, 412, 49, -796, -558, -215, 348, 57, 456, -191, 190, -137, -290, -354, 406, 293, 676, -782, -654, 475, -573, 79, -6, -76, 908, -68, -279, -406, 200, 432, 65, 331, 167, -756, 286, -616, 398, 794, 301, -448, 396]

3.14 (способ 1)(3.14.2) - сформируйте вектор $\left(x_1+2x_2-x_3, x_2+2x_3-x_4, \dots, x_{n-2}+2x_{n-1}-x_n\right)$

```
In [35]: 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, 372, 30, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Первый способ вектора:

[1052, 701, 1210, 1884, 390, 723, 1492, 1283, 394, 982, 1597, 1056, 361, 1207, 423, 443, 231, 553, 1311, 129, 914, 307, 999, 1079, 593, 325, 1831, 933, 1124, 1048, -86, 1227, 1435, 353, -475, 913, 934, 1753, 516, 757, 598, 1995, 903, -234, 1938, 980, 1061, 1626, 1716, 857, 371, 797, 1442, 1196, 485, -562, 797, 2525, 736, 1245, 852, 162, 1372, 1520, 1731, 1339, 291, -509, 1761, 902, 1206, 1410, 2220, 242, 1363, 272, 398, 2084, 845, 1444, 2108, 2000, 798, 1370, 1594, 1203, -179, 932, 1519, 624, 528, 1739, 1645, 247, 1008, -337, 641, 2422, 1232, -99, 1693, 890, 924, 1428, 1418, 50, 1536, 1242, -214, 292, 1339, 1945, 1600, 1215, 1429, 1945, 1883, 861, 406, 1204, 1009, 540, 537, 476, 1173, 257, 1895, 718, 1249, 436, 1530, 1929, 1010, 1396, -33, 1644, 167, 1807, 617, 1387, 1733, 678, 1509, 1264, 196, 769, 2247, 455, 561, 1178, 205, 2131, 1247, 12, 1073, 753, -411, 2044, 515, 904, -221, 1121, 1697, 140, 699, 409, 842, 224, 1618, 2207, 54, 749, 1962, 1565, 2074, 444, 1536, 1487, 2001, 825, -485, 1164, 1352, 1771, 1407, 1991, 819, 9, 895, 1491, 802, -333, 641, 423, 1659, 754, -379, 1570, 1047, 1179, 1625, 1338, 1883, 1231, 1203, 1406, 2268, 442, -29, 1200, 2121, 1665, 919, 979, 413, 793, 557, 1101, 1402, 1814, 1108, 1539, 633, -591, 1272, 1750, 797, 1955, 811, 284, 1953, 413, 305, 759, 2058, 366, 998, 672, -22, 693, 1945, 500, 1548, 748, -52, -133, 1980, 525]

3.14 (способ 2)(3.14.2) - сформируйте вектор $\$(x_1+2x_2-x_3, x_2+2x_3-x_4, \dots, x_{n-2}+2x_{n-1}-x_n)\$$

```
In [36]: 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, 372, 30, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Второй способ вектора:

[1052, 701, 1210, 1884, 390, 723, 1492, 1283, 394, 982, 1597, 1056, 361, 1207, 423, 443, 231, 553, 1311, 129, 914, 307, 999, 1079, 593, 325, 1831, 933, 1124, 1048, -86, 1227, 1435, 353, -475, 913, 934, 1753, 516, 757, 598, 1995, 903, -234, 1938, 980, 1061, 1626, 1716, 857, 371, 797, 1442, 1196, 485, -562, 797, 2525, 736, 1245, 852, 162, 1372, 1520, 1731, 1339, 291, -509, 1761, 902, 1206, 1410, 2220, 242, 1363, 272, 398, 2084, 845, 1444, 2108, 2000, 798, 1370, 1594, 1203, -179, 932, 1519, 624, 528, 1739, 1645, 247, 1008, -337, 641, 2422, 1232, -99, 1693, 890, 924, 1428, 1418, 50, 1536, 1242, -214, 292, 1339, 1945, 1600, 1215, 1429, 1945, 1883, 861, 406, 1204, 1009, 540, 537, 476, 1173, 257, 1895, 718, 1249, 436, 1530, 1929, 1010, 1396, -33, 1644, 167, 1807, 617, 1387, 1733, 678, 1509, 1264, 196, 769, 2247, 455, 561, 1178, 205, 2131, 1247, 12, 1073, 753, -411, 2044, 515, 904, -221, 1121, 1697, 140, 699, 409, 842, 224, 1618, 2207, 54, 749, 1962, 1565, 2074, 444, 1536, 1487, 2001, 825, -485, 1164, 1352, 1771, 1407, 1991, 819, 9, 895, 1491, 802, -333, 641, 423, 1659, 754, -379, 1570, 1047, 1179, 1625, 1338, 1883, 1231, 1203, 1406, 2268, 442, -29, 1200, 2121, 1665, 919, 979, 413, 793, 557, 1101, 1402, 1814, 1108, 1539, 633, -591, 1272, 1750, 797, 1955, 811, 284, 1953, 413, 305, 759, 2058, 366, 998, 672, -22, 693, 1945, 500, 1548, 748, -52, -133, 1980, 525]

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

```
In [37]: 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, 32 1, 157, 721, 372, 30, 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, 84 7, 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, 68 9, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 67 9, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 24 4, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 7 79, 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, 72 3, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 83 6, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 9 3, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 70 0, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 54 6, 437, 908, 621, 286, 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, 7 10, 611, 151, 0, 144, 902, 571, 957, 288, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 23 4, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 54 5, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 59 0, 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, 263, 438, 249, 113, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 6 27, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 933, 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, 46 8, 887, 609, 394, 408]

Первый способ вектора: [1.2867719843163035, -1.1454738309664125, -0.1288984082344 5167, -1.2829533770777193, -12.96493437853732, 0.9531983276634517, -0.11044491890 954632, -9.742024309355365, 1.170118876399959, -3.287544486842505, 1.385859658401 9045, 0.8275930720435408, -0.6023526590132967, 1.0047186993933188, -1.21402042830 06932, 3.123017888591526, -0.6013619125427613, -2.270052546500484, -0.47670168378 140815, 3.3653169650985846, 0.05440921729995269, 2.909684001410264, 0.13242130372 770972, -1.4875498748865863, -1.2963431176280136, -1.0807983093350686, 0.05634755 616583091, -5.63371575832417, 0.5799485547485174, 0.5419216442768281, -0.59677550 38924624, -69.83022305211773, -0.2894538138779738, -5.575562803214835, 0.12814562 710196137, 0.33257075842588635, -1.5605474772279417, -0.5344264143708287, 6.81327 8295672189, -0.9590871301529229, 0.4680191162803129, 13.046072481275216, -0.84583 72299689535, 0.6862080248863207, 0.817080881804438, 0.11471988820984633, 0.587086 1228060275, 0.21249657015284212, 0.6805875505790745, -0.7556774221733503, -0.7343 977295654383, -1.5576578224555626, 0.26389201062952317, -214.4222085295697, -1.72 91028972212166, -0.4139855502977789, 1.1926335923910525, -1.0076023656750315, -1. 0869777453840765, 0.5482942621201379, 3.9798071923904437, 0.03875125905617745, - 2.4127144812238153, 0.00014214863660449913, 228.89242117150278, 0.485797751077781 84, -0.0, -0.6898347882224115, 3.4914422053273855, -1.064893952453241, 0.93294449 75104125, -1.5835296228930222, 1.1510359298218504, -0.6981926076520347, 0.4780410 681272226, 1.05445023348073, -1.6025211350996351, -0.6992818423385311, 1.67439554 90344085, -1.121032198384096, -1.049554074743916, 0.8512030095777967, 0.363943511 27992136, -3.0207514891207747, -0.18843087346263548, 1.0051488270679492, -2.60047 2565325579, 6.4806254544222, -2.407631324782686, 0.8008441314109087, 0.2578986162 595681, 1.0801672373001179, -2.627605741675436, 0.05457370268825186, 2.4553415599 65681, -1.4464229814894676, 4.003566380855466, -0.3050702772638271, 0.98948500369 60222, 1.1406450084352604, 10.94648273676093, -1.2926176435561487, -1.87890546574 02493, -0.3406289455456973, 0.5231353475894812, 4.323365400127522, 0.851470643191 4256, 1.434824646585177, 0.9269524309326171, -1.1488785342570642, -1.803523550075

6748, 1.0658081297025837, -0.22568525906150105, -0.48518438926310736, -0.94639401
 66236955, 2.77909029770269, 0.5885850306751857, -1.5228468053986455, -0.438091400
 42990585, -0.4901913151371913, -2.313708341495891, -1.1043890162284462, -1.003761
 856900198, -1.4464434145763732, 1.1067289426340323, -0.8445878226408926, -1.42955
 19845318327, -1.4493786012392513, 1.0865556204515487, 3.517927373972851, 1.012488
 9437133153, 1.3735902012455097, 3.049668728963472, -4.3687639232597215, 1.0365048
 367806013, -0.13910076359347368, -0.7192085302391911, -1.1972409516617524, 1.1472
 115553157236, -1.3309119914597045, 1.052463610553982, -0.12728723611830572, 5.589
 662884459731, 1.2857412058031141, -1.4588400941779647, -0.7585326222188447, -1.26
 71599933021762, -0.4393677895337223, -0.4265309078451636, -0.6282659765421006, 0.
 19318716571068933, 1.9192573876234877, -0.6910265990151417, -0.6302516308190647,
 -6.8210851836648265, 2.524854028530146, 0.8829482849734308, 1.1167835068962249, -
 1.4983020235855322, 0.36765255080307546, -0.6858891385690676, 0.5325992803765964,
 -1.1248227259764203, -0.767007225132102, -1.410088204582316, 0.7086357292687817,
 0.13496233821350037, 1.2713454183497739, -7.672674288850925, -1.7627730176993472,
 1.3236579893849412, -0.9626026355722461, -0.9930369906475578, 5.905253818219098,
 -0.8241496151973007, -1.3856935955686873, -0.7904173665704957, 2.522765369062866,
 -1.2049455871432027, 0.9693211292292829, -1.2446973961155001, 0.1225229416601494
 2, -1.2598683100911872, -0.6523290473328525, -0.2194434511580926, 0.9401696071617
 609, -3.5081837269892318, 0.8991368614871886, 0.3049791392496953, 14.110033961675
 793, 13.36069943097705, -1.0649230085119799, -0.46873597849956233, -0.06245615886
 2407206, 0.8080609235257259, -5.120035857885296, 1.3993091552320849, 0.1604721455
 6978037, 1.0011021773932103, 0.17923024216452463, 0.430526749137862, 2.2690590210
 404498, 0.6093024166418017, -1.41130842764286, 0.8445452353321987, 0.450547602691
 15923, -0.9849599716541627, -0.35150528554019805, -0.16738201701067565, -0.437896
 8818452856, -0.646579685314153, 13.853359471000587, -0.49650463948189527, 1.59690
 96465547614, 3.183503121600984, -0.6447283929279362, 1.7403048203849318, 0.055190
 57659651424, -0.3421730957616319, 1.3949198931041937, 1.0370755094417903, 1.40381
 57924883594, 1.4272084565243293, -1.380371254322068, -0.018833692662689954, -0.97
 58663334347849, 2.648318212546895, 0.43658272908273893, -1.641195702534428, 4.538
 344259023536, 0.9855004479471506, -0.9327954917761051, -1.3882837509634576, 0.864
 0970394561978, 0.7823937551721631, 1.3070475813798146, 0.8575495448596192, 0.0442
 9348301104583, -1.3486936039257187, 1.7532303169984271, -0.46032677032787567, 1.3
 842840602287623, -0.46116182639964265, -0.961632159666424, -0.5546934448763409,
 0.09790264677088786, 0.8788606334074082, -0.5356045366141032, 7.047042520084093]

3.14 (способ 2)(3.14.3) - сформируйте вектор $\left(\frac{\sin \left(y_1\right)}{\cos \left(x_2\right)}, \frac{\sin \left(y_2\right)}{\cos \left(x_3\right)}, \ldots, \frac{\sin \left(y_{n-1}\right)}{\cos \left(x_n\right)}\right)$;

```
In [38]: 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, 32 1, 157, 721, 372, 30, 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, 84 7, 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, 68 9, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 67 9, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 24 4, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 7 79, 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, 72 3, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 83 6, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 9 3, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 70 0, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 54 6, 437, 908, 621, 286, 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, 7 10, 611, 151, 0, 144, 902, 571, 957, 288, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 23 4, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 54 5, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 59 0, 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, 263, 438, 249, 113, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 6 27, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 933, 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, 46 8, 887, 609, 394, 408]

Второй способ вектора: [1.2867719843163035, -1.1454738309664125, -0.1288984082344 5167, -1.2829533770777193, -12.96493437853732, 0.9531983276634517, -0.11044491890 954632, -9.742024309355365, 1.170118876399959, -3.287544486842505, 1.385859658401 9045, 0.8275930720435408, -0.6023526590132967, 1.0047186993933188, -1.21402042830 06932, 3.123017888591526, -0.6013619125427613, -2.270052546500484, -0.47670168378 140815, 3.3653169650985846, 0.05440921729995269, 2.909684001410264, 0.13242130372 770972, -1.4875498748865863, -1.2963431176280136, -1.0807983093350686, 0.05634755 616583091, -5.63371575832417, 0.5799485547485174, 0.5419216442768281, -0.59677550 38924624, -69.83022305211773, -0.2894538138779738, -5.575562803214835, 0.12814562 710196137, 0.33257075842588635, -1.5605474772279417, -0.5344264143708287, 6.81327 8295672189, -0.9590871301529229, 0.4680191162803129, 13.046072481275216, -0.84583 72299689535, 0.6862080248863207, 0.817080881804438, 0.11471988820984633, 0.587086 1228060275, 0.21249657015284212, 0.6805875505790745, -0.7556774221733503, -0.7343 977295654383, -1.5576578224555626, 0.26389201062952317, -214.4222085295697, -1.72 91028972212166, -0.4139855502977789, 1.1926335923910525, -1.0076023656750315, -1. 0869777453840765, 0.5482942621201379, 3.9798071923904437, 0.03875125905617745, - 2.4127144812238153, 0.00014214863660449913, 228.89242117150278, 0.485797751077781 84, -0.0, -0.6898347882224115, 3.4914422053273855, -1.064893952453241, 0.93294449 75104125, -1.5835296228930222, 1.1510359298218504, -0.6981926076520347, 0.4780410 681272226, 1.05445023348073, -1.6025211350996351, -0.6992818423385311, 1.67439554 90344085, -1.121032198384096, -1.049554074743916, 0.8512030095777967, 0.363943511 27992136, -3.0207514891207747, -0.18843087346263548, 1.0051488270679492, -2.60047 2565325579, 6.4806254544222, -2.407631324782686, 0.8008441314109087, 0.2578986162 595681, 1.0801672373001179, -2.627605741675436, 0.05457370268825186, 2.4553415599 65681, -1.4464229814894676, 4.003566380855466, -0.3050702772638271, 0.98948500369 60222, 1.1406450084352604, 10.94648273676093, -1.2926176435561487, -1.87890546574 02493, -0.3406289455456973, 0.5231353475894812, 4.323365400127522, 0.851470643191 4256, 1.434824646585177, 0.9269524309326171, -1.1488785342570642, -1.803523550075

6748, 1.0658081297025837, -0.22568525906150105, -0.48518438926310736, -0.94639401
 66236955, 2.77909029770269, 0.5885850306751857, -1.5228468053986455, -0.438091400
 42990585, -0.4901913151371913, -2.313708341495891, -1.1043890162284462, -1.003761
 856900198, -1.4464434145763732, 1.1067289426340323, -0.8445878226408926, -1.42955
 19845318327, -1.4493786012392513, 1.0865556204515487, 3.517927373972851, 1.012488
 9437133153, 1.3735902012455097, 3.049668728963472, -4.3687639232597215, 1.0365048
 367806013, -0.13910076359347368, -0.7192085302391911, -1.1972409516617524, 1.1472
 115553157236, -1.3309119914597045, 1.052463610553982, -0.12728723611830572, 5.589
 662884459731, 1.2857412058031141, -1.4588400941779647, -0.7585326222188447, -1.26
 71599933021762, -0.4393677895337223, -0.4265309078451636, -0.6282659765421006, 0.
 19318716571068933, 1.9192573876234877, -0.6910265990151417, -0.6302516308190647,
 -6.8210851836648265, 2.524854028530146, 0.8829482849734308, 1.1167835068962249, -
 1.4983020235855322, 0.36765255080307546, -0.6858891385690676, 0.5325992803765964,
 -1.1248227259764203, -0.767007225132102, -1.410088204582316, 0.7086357292687817,
 0.13496233821350037, 1.2713454183497739, -7.672674288850925, -1.7627730176993472,
 1.3236579893849412, -0.9626026355722461, -0.9930369906475578, 5.905253818219098,
 -0.8241496151973007, -1.3856935955686873, -0.7904173665704957, 2.522765369062866,
 -1.2049455871432027, 0.9693211292292829, -1.2446973961155001, 0.1225229416601494
 2, -1.2598683100911872, -0.6523290473328525, -0.2194434511580926, 0.9401696071617
 609, -3.5081837269892318, 0.8991368614871886, 0.3049791392496953, 14.110033961675
 793, 13.36069943097705, -1.0649230085119799, -0.46873597849956233, -0.06245615886
 2407206, 0.8080609235257259, -5.120035857885296, 1.3993091552320849, 0.1604721455
 6978037, 1.0011021773932103, 0.17923024216452463, 0.430526749137862, 2.2690590210
 404498, 0.6093024166418017, -1.41130842764286, 0.8445452353321987, 0.450547602691
 15923, -0.9849599716541627, -0.35150528554019805, -0.16738201701067565, -0.437896
 8818452856, -0.646579685314153, 13.853359471000587, -0.49650463948189527, 1.59690
 96465547614, 3.183503121600984, -0.6447283929279362, 1.7403048203849318, 0.055190
 57659651424, -0.3421730957616319, 1.3949198931041937, 1.0370755094417903, 1.40381
 57924883594, 1.4272084565243293, -1.380371254322068, -0.018833692662689954, -0.97
 58663334347849, 2.648318212546895, 0.43658272908273893, -1.641195702534428, 4.538
 344259023536, 0.9855004479471506, -0.9327954917761051, -1.3882837509634576, 0.864
 0970394561978, 0.7823937551721631, 1.3070475813798146, 0.8575495448596192, 0.0442
 9348301104583, -1.3486936039257187, 1.7532303169984271, -0.46032677032787567, 1.3
 842840602287623, -0.46116182639964265, -0.961632159666424, -0.5546934448763409,
 0.09790264677088786, 0.8788606334074082, -0.5356045366141032, 7.047042520084093]

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

In [39]: `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, 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, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Первый способ суммы:

0.00015970776348493994

3.14 (способ 2)(3.14.4) - вычислите $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$;

```
In [40]: 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, 2
72, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 72
1, 372, 30, 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, 69
9, 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, 673, 82
0, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 94
3, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 244, 673, 82
1, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 4
8, 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, 20
5, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384,
416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 42
9, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 84
2, 12, 341]
```

Вектор y:

```
[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 89
9, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 90
8, 621, 286, 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, 1
51, 0, 144, 902, 571, 957, 288, 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, 96
8, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950,
859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 77
8, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 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, 263, 438, 249, 11
3, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 11
3, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 16
9, 764, 294, 933, 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, 60
9, 394, 408]
```

Второй способ суммы:

```
0.0001597077634849399
```

3.14 (способ 1)(3.14.5) - выберите элементы вектора \$y\$, значения которых больше 600 , и выведите на экран; определите индексы этих элементов;

```
In [41]: 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, 712, 904, 717, 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, 644, 887, 673, 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, 107, 108, 110, 111, 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, 196, 197, 200, 208, 210, 211, 212, 216, 218, 220, 224, 226, 229, 233, 234, 239, 242, 247, 248]

3.14 (способ 2)(3.14.5) - выберите элементы вектора y , значения которых больше 600, и выведите на экран; определите индексы этих элементов;

```
In [42]: 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, 712, 904, 717, 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, 644, 887, 673, 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, 107, 108, 110, 111, 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, 196, 197, 200, 208, 210, 211, 212, 216, 218, 220, 224, 226, 229, 233, 234, 239, 242, 247, 248]

3.14 (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);

```
In [43]: 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)
```

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, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Значения вектора x, соответствующие $y > 600$ (первый способ):

[703, 165, 457, 299, 515, 90, 272, 191, 484, 196, 315, 319, 721, 372, 713, 165, 527, 296, 999, 731, 619, 279, 320, 111, 81, 835, 218, 281, 847, 686, 194, 359, 711, 695, 435, 587, 86, 702, 299, 835, 949, 184, 798, 44, 546, 689, 673, 820, 359, 306, 915, 350, 943, 426, 126, 628, 398, 746, 821, 511, 346, 173, 545, 172, 466, 779, 227, 484, 799, 942, 819, 285, 867, 748, 227, 484, 331, 723, 118, 449, 779, 399, 858, 915, 237, 398, 729, 84, 909, 836, 28, 419, 169, 769, 308, 842]

3.14 (способ 2)(3.14.6) - определите значения вектора x , соответствующие значениям вектора y , значения которых больше 600 (под соответствием понимается расположение на аналогичных индексных позициях);

```
In [44]: using Random

Random.seed!(42)

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

selected_x2 = [x[i] for i in findall(y -> y > 600, y)]

println("Вектор x:")
```

```
println(x)

println("\nВектор y:")
println(y)

println("\nЗначения вектора x, соответствующие y > 600 (второй способ):")
println(selected_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, 157, 721, 372, 30, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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]

Значения вектора x, соответствующие y > 600 (второй способ):

[703, 165, 457, 299, 515, 90, 272, 191, 484, 196, 315, 319, 721, 372, 713, 165, 527, 296, 999, 731, 619, 279, 320, 111, 81, 835, 218, 281, 847, 686, 194, 359, 711, 695, 435, 587, 86, 702, 299, 835, 949, 184, 798, 44, 546, 689, 673, 820, 359, 306, 915, 350, 943, 426, 126, 628, 398, 746, 821, 511, 346, 173, 545, 172, 466, 779, 227, 484, 799, 942, 819, 285, 867, 748, 227, 484, 331, 723, 118, 449, 779, 399, 858, 915, 237, 398, 729, 84, 909, 836, 28, 419, 169, 769, 308, 842]

3.14 (способ 1)(3.14.7) - сформируйте вектор $\left(\left|x_1 - \bar{x}\right|^{\frac{1}{2}}, \left|x_2 - \bar{x}\right|^{\frac{1}{2}}, \dots, \left|x_n - \bar{x}\right|^{\frac{1}{2}}\right)$, где \bar{x} обозначает среднее значение вектора $x = (x_1, x_2, \dots, x_n)$;

In [45]: `using Random`


```
Random.seed!(42)

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

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

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

495.472

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 2
72, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 72
1, 372, 30, 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, 69
9, 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, 673, 82
0, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 94
3, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 244, 673, 82
1, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 4
8, 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, 20
5, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384,
416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 42
9, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 84
2, 12, 341]

Вектор y:

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 89
9, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 90
8, 621, 286, 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, 1
51, 0, 144, 902, 571, 957, 288, 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, 96
8, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950,
859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 77
8, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 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, 263, 438, 249, 11
3, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 11
3, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 16
9, 764, 294, 933, 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, 60
9, 394, 408]

Среднее значение вектора x: 495.472

Вектор:

[11.555431623267044, 6.743292964123684, 4.297906467106977, 14.405832152291655, 1
3.323963374311715, 18.1788888549328, 10.84103316109678, 13.134991435094276, 6.202
580108309765, 14.016847006370583, 12.865768535147833, 11.980317191126453, 12.3883
81653791587, 15.115290271774471, 4.4190496715923, 20.136335317033236, 14.94897989
8307442, 17.449126052613636, 8.513048807565946, 3.3870341008026448, 20.3831302797
19058, 4.4190496715923, 17.30525931617322, 10.22389358317075, 9.923305900757065,
13.433986750030684, 7.840408152641033, 19.040168066485126, 13.284276419888288, 8.
217542211634816, 13.208784955475654, 18.3976085402424, 15.017589686763985, 11.111
795534475965, 21.57480011494892, 20.40764562608828, 12.943260794714755, 1.8633303
51816333, 14.748830462107835, 18.1788888549328, 5.614979964345378, 5.785499114164
653, 18.90841082693096, 17.95750539468108, 14.123455667788956, 22.43942958276791
8, 11.809826417013927, 15.346921515404972, 16.17182735500228, 11.114315093607884,
14.712987460063982, 13.246584465438628, 7.247620299105081, 11.114315093607884, 1
2.307396150282967, 19.607957568293543, 20.3585854125477, 18.426285572518406, 21.4
13266915629666, 16.657490807441558, 12.59079028496623, 14.644862580440964, 11.466
124018167603, 18.749079977428227, 13.803188037551326, 14.26632398342334, 11.93616
353775366, 20.723706232235585, 17.362949058267724, 21.6223957969509, 11.682123094
71185, 16.838289699372677, 14.68087190871169, 17.190927839997467, 20.554123673851
922, 14.125438046304971, 19.88647781785402, 11.292829583412654, 21.2727055166943
9, 7.776374476579686, 21.852414054287, 21.01256766794577, 15.605383686407714, 8.9

70618707759236, 16.59903611659424, 9.567026706349262, 12.01965057728385, 20.235414500325906, 14.371082074777808, 7.907464827617004, 14.016847006370583, 6.043839839042727, 18.426285572518406, 5.698420833880205, 19.480041067718517, 6.889992743102128, 22.16916777869661, 17.248999971012815, 21.296196843568104, 15.048986676849706, 15.507159636761337, 19.558323036497786, 13.321861731755062, 11.337019008540121, 12.10487505098669, 3.9334463260606443, 17.648569347117064, 17.393332055704565, 15.857868709255982, 21.247870481532967, 7.108305001897429, 18.66890462774932, 19.988196516944694, 13.911434146054102, 13.323963374311715, 18.014660696221842, 19.71113616434766, 12.146110488547354, 14.438559484934776, 11.809826417013927, 9.875626562400992, 12.548784801724826, 15.345096936806883, 13.321861731755062, 11.68212309471185, 8.032932216818464, 13.764882854568722, 20.482382673898073, 15.952178534607741, 13.547250643580785, 12.061177388630016, 21.154857598197157, 14.509583040184168, 8.33498650268853, 7.247620299105081, 19.2216544553272, 18.371935118544265, 18.640600848685107, 21.59925924655751, 15.181304291792586, 19.481478383326046, 11.512080611253555, 9.872790892143922, 15.828076320260779, 10.699158845442009, 15.857868709255982, 13.323963374311715, 18.04239451957528, 20.675396005880998, 2.554995107627414, 3.9405583360737118, 12.225874201872028, 22.417136302391526, 16.86629775617637, 17.95750539468108, 7.037613231771125, 17.477757293199833, 17.985327353150957, 21.202075370114127, 21.85113269375297, 5.428812024743533, 21.2007547035477, 16.838289699372677, 3.5315718879841564, 21.153533983710616, 7.514785426078378, 16.385115196421413, 3.3870341008026448, 11.93616353775366, 21.691657382505376, 13.096869855045519, 20.06170481290162, 17.422054987859497, 21.13120914666267, 15.017589686763985, 17.986884110373314, 14.50765315273287, 21.202075370114127, 1.980317191126453, 15.50896514922901, 19.633440859920604, 19.09114978203251, 19.013889659930186, 13.620866345427519, 19.27506160820245, 12.187206406720124, 15.891129601132828, 18.613758352358612, 16.385115196421413, 9.92612714002798, 4.95257508777, 18.7742376676125, 22.16916777869661, 3.3870341008026448, 12.824663738281794, 15.084031291402177, 19.428638655345875, 17.043239128757186, 20.286152912762933, 6.817037479726804, 16.202715821738035, 17.044881929775872, 14.47508203776407, 19.040168066485126, 6.5213495535816834, 14.405832152291655, 15.6693331064216, 16.838289699372677, 21.55161246867621, 9.822016086323622, 19.040168066485126, 20.482382673898073, 8.457422775290356, 10.558030119297822, 8.914706949754432, 16.077064408653715, 4.297906467106977, 9.872790892143922, 14.850185184030535, 15.281622950459157, 16.62913106569312, 6.039205245725631, 9.302042786399126, 20.28477261395848, 19.454356838507923, 20.33538787434358, 12.943260794714755, 3.2360469712289377, 18.453400770589685, 17.131024487753205, 8.153036244246678, 16.748970117592307, 21.621100804538145, 8.744827042314787, 8.094936689066815, 16.927137974270785, 20.699565212825124, 7.38430768589717, 18.06853618863465, 16.717416068280407, 11.292829583412654, 16.53868193055299, 16.717416068280407, 14.336247765716106, 20.62697263293865, 20.06170481290162, 13.69204148401545, 18.61526255522602, 21.987996725486386, 12.42867651843912]

3.14 (способ 2)(3.14.7) - сформируйте вектор $\left(\left|x_1 - \bar{x}\right|^{\frac{1}{2}}, \left|x_2 - \bar{x}\right|^{\frac{1}{2}}, \dots, \left|x_n - \bar{x}\right|^{\frac{1}{2}}\right)$, где \bar{x} обозначает среднее значение вектора $x = (x_1, x_2, \dots, x_n)$;

```
In [46]: 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("\nВектор y:")
println(y)
println("\nСреднее значение вектора x: ", mean_x)
println("\nВектор result_vector:")
println(result_vector)
```

Вектор x:

[629, 450, 477, 703, 673, 165, 613, 668, 457, 299, 661, 639, 342, 267, 515, 90, 2
72, 191, 423, 484, 80, 515, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 72
1, 372, 30, 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, 69
9, 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, 673, 82
0, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 679, 350, 94
3, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 244, 673, 82
1, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 779, 483, 4
8, 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, 20
5, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384,
416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 836, 202, 42
9, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 93, 308, 84
2, 12, 341]

Вектор y:

[551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 700, 973, 89
9, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 546, 437, 90
8, 621, 286, 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, 1
51, 0, 144, 902, 571, 957, 288, 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, 96
8, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950,
859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 545, 888, 77
8, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 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, 263, 438, 249, 11
3, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 627, 872, 11
3, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 16
9, 764, 294, 933, 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, 60
9, 394, 408]

Среднее значение вектора x: 495.472

Вектор result_vector:

[11.555431623267044, 6.743292964123684, 4.297906467106977, 14.405832152291655, 1
3.323963374311715, 18.1788888549328, 10.84103316109678, 13.134991435094276, 6.202
580108309765, 14.016847006370583, 12.865768535147833, 11.980317191126453, 12.3883
81653791587, 15.115290271774471, 4.4190496715923, 20.136335317033236, 14.94897989
8307442, 17.449126052613636, 8.513048807565946, 3.3870341008026448, 20.3831302797
19058, 4.4190496715923, 17.30525931617322, 10.22389358317075, 9.923305900757065,
13.433986750030684, 7.840408152641033, 19.040168066485126, 13.284276419888288, 8.
217542211634816, 13.208784955475654, 18.3976085402424, 15.017589686763985, 11.111
795534475965, 21.57480011494892, 20.40764562608828, 12.943260794714755, 1.8633303
51816333, 14.748830462107835, 18.1788888549328, 5.614979964345378, 5.785499114164
653, 18.90841082693096, 17.95750539468108, 14.123455667788956, 22.43942958276791
8, 11.809826417013927, 15.346921515404972, 16.17182735500228, 11.114315093607884,
14.712987460063982, 13.246584465438628, 7.247620299105081, 11.114315093607884, 1
2.307396150282967, 19.607957568293543, 20.3585854125477, 18.426285572518406, 21.4
13266915629666, 16.657490807441558, 12.59079028496623, 14.644862580440964, 11.466
124018167603, 18.749079977428227, 13.803188037551326, 14.26632398342334, 11.93616
353775366, 20.723706232235585, 17.362949058267724, 21.6223957969509, 11.682123094
71185, 16.838289699372677, 14.68087190871169, 17.190927839997467, 20.554123673851
922, 14.125438046304971, 19.88647781785402, 11.292829583412654, 21.2727055166943
9, 7.776374476579686, 21.852414054287, 21.01256766794577, 15.605383686407714, 8.9
70618707759236, 16.59903611659424, 9.567026706349262, 12.01965057728385, 20.23541

4500325906, 14.371082074777808, 7.907464827617004, 14.016847006370583, 6.04383983
 9042727, 18.426285572518406, 5.698420833880205, 19.480041067718517, 6.88999274310
 2128, 22.16916777869661, 17.248999971012815, 21.296196843568104, 15.0489866768497
 06, 15.507159636761337, 19.558323036497786, 13.321861731755062, 11.33701900854012
 1, 12.10487505098669, 3.9334463260606443, 17.648569347117064, 17.393332055704565,
 15.857868709255982, 21.247870481532967, 7.108305001897429, 18.66890462774932, 19.
 988196516944694, 13.911434146054102, 13.323963374311715, 18.014660696221842, 19.7
 11113616434766, 12.146110488547354, 14.438559484934776, 11.809826417013927, 9.875
 626562400992, 12.548784801724826, 15.345096936806883, 13.321861731755062, 11.6821
 2309471185, 8.032932216818464, 13.764882854568722, 20.482382673898073, 15.9521785
 34607741, 13.547250643580785, 12.061177388630016, 21.154857598197157, 14.50958304
 0184168, 8.33498650268853, 7.247620299105081, 19.2216544553272, 18.37193511854426
 5, 18.640600848685107, 21.59925924655751, 15.181304291792586, 19.481478383326046,
 11.512080611253555, 9.872790892143922, 15.828076320260779, 10.699158845442009, 1
 5.857868709255982, 13.323963374311715, 18.04239451957528, 20.675396005880998, 2.5
 54995107627414, 3.9405583360737118, 12.225874201872028, 22.417136302391526, 16.86
 629775617637, 17.95750539468108, 7.037613231771125, 17.477757293199833, 17.985327
 353150957, 21.202075370114127, 21.85113269375297, 5.428812024743533, 21.200754703
 5477, 16.838289699372677, 3.5315718879841564, 21.153533983710616, 7.5147854260783
 78, 16.385115196421413, 3.3870341008026448, 11.93616353775366, 21.69165738250537
 6, 13.096869855045519, 20.06170481290162, 17.422054987859497, 21.13120914666267,
 15.017589686763985, 17.986884110373314, 14.50765315273287, 21.202075370114127, 1
 1.980317191126453, 15.50896514922901, 19.633440859920604, 19.09114978203251, 19.0
 13889659930186, 13.620866345427519, 19.27506160820245, 12.187206406720124, 15.891
 129601132828, 18.613758352358612, 16.385115196421413, 9.92612714002798, 4.9525750
 8777, 18.7742376676125, 22.16916777869661, 3.3870341008026448, 12.82466373828179
 4, 15.084031291402177, 19.428638655345875, 17.043239128757186, 20.28615291276293
 3, 6.817037479726804, 16.202715821738035, 17.044881929775872, 14.47508203776407,
 19.040168066485126, 6.5213495535816834, 14.405832152291655, 15.6693331064216, 16.
 838289699372677, 21.55161246867621, 9.822016086323622, 19.040168066485126, 20.482
 382673898073, 8.457422775290356, 10.558030119297822, 8.914706949754432, 16.077064
 408653715, 4.297906467106977, 9.872790892143922, 14.850185184030535, 15.281622950
 459157, 16.62913106569312, 6.039205245725631, 9.302042786399126, 20.2847726139584
 8, 19.454356838507923, 20.33538787434358, 12.943260794714755, 3.2360469712289377,
 18.453400770589685, 17.131024487753205, 8.153036244246678, 16.748970117592307, 2
 1.621100804538145, 8.744827042314787, 8.094936689066815, 16.927137974270785, 20.6
 99565212825124, 7.38430768589717, 18.06853618863465, 16.717416068280407, 11.29282
 9583412654, 16.53868193055299, 16.717416068280407, 14.336247765716106, 20.6269726
 3293865, 20.06170481290162, 13.69204148401545, 18.61526255522602, 21.987996725486
 386, 12.42867651843912]

3.14 (способ 1)(3.14.8) - определите, сколько элементов вектора \$y\$ отстоят от
 максимального значения не более, чем на 200 ;

```
In [47]: 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
```

Вектор у: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 70, 0, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 54, 6, 437, 908, 621, 286, 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, 7, 10, 611, 151, 0, 144, 902, 571, 957, 288, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 23, 4, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 54, 5, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 59, 0, 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, 263, 438, 249, 113, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 6, 27, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 933, 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, 46, 8, 887, 609, 394, 408]

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

3.14 (способ 2)(3.14.8) - определите, сколько элементов вектора y отстоят от максимального значения не более, чем на 200 ;

```
In [48]: 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
```

Вектор у: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 70, 0, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 54, 6, 437, 908, 621, 286, 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, 7, 10, 611, 151, 0, 144, 902, 571, 957, 288, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 23, 4, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 54, 5, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 59, 0, 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, 263, 438, 249, 113, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 6, 27, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 933, 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, 46, 8, 887, 609, 394, 408]

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

3.14 (способ 1)(3.14.9) - определите, сколько чётных и нечётных элементов вектора \$x\$;

In [49]:

```
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, 32, 1, 157, 721, 372, 30, 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, 84, 7, 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, 68, 9, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 67, 9, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 24, 4, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 7, 79, 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, 72, 3, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 83, 6, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 9, 3, 308, 842, 12, 341]

Четные элементы в x: 112

Нечетные элементы в x: 138

3.14 (способ 2)(3.14.9) - определите, сколько чётных и нечётных элементов вектора \$x\$;


```
In [50]: 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, 32, 1, 157, 721, 372, 30, 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, 84, 7, 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, 68, 9, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 67, 9, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 24, 4, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 7, 79, 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, 72, 3, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 83, 6, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 9, 3, 308, 842, 12, 341]

Четные элементы в x: 112

Нечетные элементы в x: 138

3.14 (способ 1)(3.14.10) - определите, сколько элементов вектора x кратны 7;

```
In [51]: 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, 157, 721, 372, 30, 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, 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, 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, 398, 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

3.14 (способ 2)(3.14.10) - определите, сколько элементов вектора x кратны 7;

In [52]: `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, 157, 721, 372, 30, 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, 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, 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, 398, 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

3.14 (способ 1)(3.14.11) - отсортируйте элементы вектора x в порядке возрастания элементов вектора y ;

In [53]: `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):
```

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, 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, 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, 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, 398, 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, 437, 908, 621, 286, 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, 957, 288, 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, 130, 755, 937, 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, 889, 915, 753, 439, 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, 263, 438, 249, 113, 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, 764, 294, 933, 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 1): [353, 48, 668, 384, 485, 216, 667, 492, 46, 661, 532, 643, 538, 673, 31, 853, 70, 255, 116, 149, 907, 582, 948, 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, 973, 110, 356, 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, 100, 67, 117, 786, 793, 12, 260, 520, 341, 663, 450, 593, 937, 353, 776, 644, 833, 241, 157, 736, 381, 415, 558, 654, 269, 190, 93, 318, 858, 79, 459, 619, 857, 706, 73, 483, 439, 701, 364, 548, 758, 600, 721, 998, 342, 594, 679, 321, 623, 356, 629, 423, 963, 434, 463, 858, 211, 563, 244, 772, 613, 713, 842, 686, 191, 372, 511, 723, 315, 346, 915, 484, 942, 299, 426, 227, 285, 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, 218, 820, 999, 81, 867, 118, 731, 308, 819, 350, 628, 272, 194, 435, 729, 915, 721, 165, 799, 398, 457, 86, 398, 821, 419, 798, 484, 673, 748, 296, 359, 836, 779, 320, 111, 835, 949, 90, 779, 711, 281, 331, 169, 306, 84, 165, 359]

3.14 (способ 2)(3.14.11) - отсортируйте элементы вектора x в порядке возрастания элементов вектора y ;

```
In [54]: 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):

```

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, 32, 1, 157, 721, 372, 30, 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, 84, 7, 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, 68, 9, 673, 820, 884, 643, 287, 356, 593, 338, 260, 318, 359, 560, 306, 915, 241, 67, 9, 350, 943, 706, 426, 548, 126, 833, 148, 962, 265, 875, 628, 398, 746, 381, 24, 4, 673, 821, 68, 502, 511, 346, 998, 211, 173, 545, 190, 172, 945, 18, 466, 46, 7, 79, 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, 72, 3, 118, 205, 907, 449, 758, 786, 705, 858, 538, 703, 741, 779, 31, 399, 858, 915, 567, 384, 416, 237, 477, 398, 716, 729, 772, 459, 582, 84, 117, 909, 663, 485, 83, 6, 202, 429, 776, 28, 419, 561, 782, 67, 550, 169, 216, 623, 769, 216, 701, 70, 9, 3, 308, 842, 12, 341]

Вектор y: [551, 413, 113, 828, 266, 909, 600, 8, 922, 740, 35, 337, 524, 140, 70, 0, 973, 899, 611, 565, 946, 132, 146, 735, 519, 111, 627, 578, 582, 833, 587, 54, 6, 437, 908, 621, 286, 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, 7, 10, 611, 151, 0, 144, 902, 571, 957, 288, 975, 277, 496, 712, 375, 255, 107, 904, 249, 414, 185, 447, 292, 717, 165, 928, 821, 450, 663, 45, 762, 578, 92, 240, 23, 4, 393, 968, 451, 87, 347, 128, 374, 364, 130, 755, 937, 247, 796, 731, 300, 308, 829, 950, 859, 295, 51, 239, 547, 414, 202, 403, 470, 995, 283, 988, 642, 435, 54, 5, 888, 778, 492, 664, 517, 757, 428, 181, 277, 135, 278, 889, 915, 753, 439, 59, 0, 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, 263, 438, 249, 113, 492, 370, 870, 423, 951, 103, 672, 536, 403, 139, 232, 644, 981, 6, 27, 872, 113, 105, 779, 518, 392, 257, 479, 59, 346, 193, 961, 63, 719, 811, 907, 119, 9, 169, 764, 294, 933, 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, 46, 8, 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, 149, 907, 582, 948, 397, 477, 131, 205, 567, 663, 318, 480, 80, 265, 143, 267, 6, 6, 515, 699, 966, 351, 416, 148, 739, 741, 338, 68, 716, 93, 954, 344, 4, 4, 216, 287, 448, 244, 18, 973, 110, 356, 623, 705, 945, 639, 202, 673, 550, 791, 962, 87, 5, 429, 757, 782, 560, 30, 779, 771, 477, 884, 844, 548, 895, 502, 173, 639, 703, 878, 561, 945, 642, 681, 462, 624, 100, 67, 117, 786, 793, 12, 260, 520, 341, 66, 3, 450, 937, 593, 353, 644, 776, 833, 241, 157, 736, 381, 415, 654, 558, 269, 19, 0, 93, 318, 858, 79, 459, 619, 706, 857, 73, 483, 439, 701, 364, 548, 758, 600, 7, 21, 342, 998, 594, 679, 321, 356, 623, 629, 423, 963, 434, 463, 858, 211, 563, 24, 4, 772, 613, 713, 842, 191, 686, 372, 511, 315, 723, 346, 915, 942, 484, 299, 42, 6, 227, 285, 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, 68, 9, 319, 527, 218, 820, 999, 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, 35, 9, 836, 320, 111, 779, 835, 949, 90, 779, 711, 281, 331, 169, 306, 84, 165, 359]

3.14 (способ 1)(3.14.12) - выведите элементы вектора x , которые входят в десятку наибольших (top-10)?

```
In [55]: 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]

3.14 (способ 2)(3.14.12) - выведите элементы вектора x , которые входят в десятку наибольших (top-10)?

```
In [56]: 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]

3.14 (способ 1)(3.14.13) - сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x

```
In [57]: 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, 2
72, 191, 423, 484, 80, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721, 37
2, 30, 79, 663, 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, 19
4, 963, 359, 779, 711, 791, 73, 695, 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, 895, 689, 820, 884, 643, 287, 593, 338, 26
0, 560, 306, 915, 241, 679, 350, 943, 706, 426, 126, 833, 148, 962, 265, 875, 62
8, 398, 746, 381, 821, 68, 502, 511, 346, 998, 211, 545, 190, 172, 945, 18, 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, 705, 5
38, 741, 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]
```

3.14 (способ 2)(3.14.13) - сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора \$x\$

In [58]:

```
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, 9
0, 272, 191, 423, 484, 80, 196, 600, 397, 315, 434, 858, 319, 563, 321, 157, 721,
372, 30, 79, 663, 492, 713, 527, 462, 853, 173, 296, 999, 356, 731, 757, 619, 27
9, 320, 548, 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, 77
1, 587, 351, 86, 702, 558, 532, 463, 116, 448, 4, 793, 949, 269, 255, 878, 318, 6
24, 642, 480, 184, 798, 244, 44, 546, 844, 895, 689, 820, 884, 643, 287, 593, 33
8, 260, 560, 306, 915, 241, 679, 350, 943, 706, 426, 126, 833, 148, 962, 265, 87
5, 628, 398, 746, 381, 821, 68, 502, 511, 346, 998, 211, 545, 190, 172, 945, 18,
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, 7
05, 538, 741, 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]
```

4. Создайте массив squares, в котором будут храниться квадраты всех целых чисел от 1 до 100.

```
In [59]: # Создаем массив squares с квадратами целых чисел от 1 до 100
squares = [i^2 for i in 1:100]

# Выводим результат
println(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]
```

5. Подключите пакет Primes (функции для вычисления простых чисел).

Сгенерируйте массив myprimes, в котором будут храниться первые 168 простых чисел. Определите 89-е наименьшее простое число. Получите срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа.

```
In [60]: # 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, 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]
```

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. Вычислите следующие выражения:

6.1) $\sum_{i=10}^{100} (i^3 + 4i^2)$;

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

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

6.1

```
In [61]: # Вычисляем сумму выражения  $\sum(i^3 + 4i^2)$  для  $i$  от 10 до 100
result = sum(i^3 + 4i^2 for i in 10:100)

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

Результат выражения: 26852735

6.2

```
In [62]: # Установим значение M
M = 25

# Вычисляем 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 1:M)

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

Результат выражения: 2.1291704368143802e9

6.3

```
In [63]: # Установим значение N
N = 20

# Вычисляем сумму выражения  $1 + \frac{2}{3} + \left(\frac{2}{3} \cdot \frac{4}{5}\right) + \dots + \left(\frac{2}{3} \cdot \frac{4}{5} \cdot \dots \cdot \frac{38}{39}\right)$ 
result = 1.0 + sum(prod([(2 * i)/(2 * i + 1) for i in 1:n]) for n in 1:N)

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

Результат выражения: 7.170891165651219

Выводы:

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

Список литературы:

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

https://esystem.rudn.ru/pluginfile.php/2231396/mod_resource/content/2/002-lab_data-types.pdf.