Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**Лабораторна робота №3**

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІС-23

Шимків Мирослав

Київ 2025

**ЛАБОРАТОРНАЯ РОБОТА №3**

**Завдання**: за допомогою мультипарадигменної мови R реалізувати перетворення чисельного ряду лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів з подальшою побудовою матриці передування.

**Вхідні данні**: чисельний ряд, вид розподілу ймовірностей, потужність алфавіту.

**Вихідні дані**: лінгвістичний ряд та матриця передування.

**Мова програмування**: R

**Варіант 24**

(завдання 4 відповідно)

Мій варіант 4, отже, я обрав Хі-розподіл і мову програмування R Lang

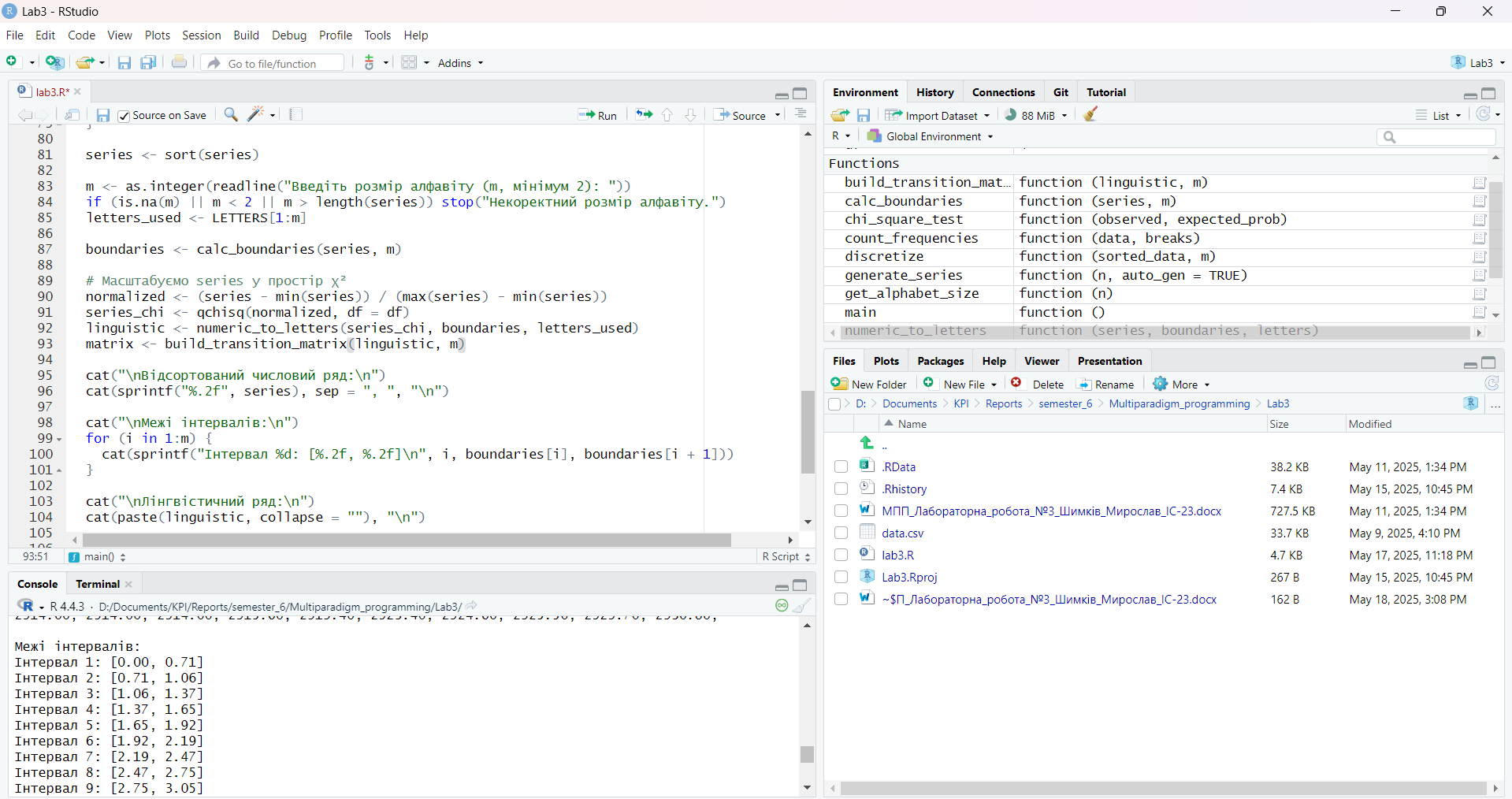
**Виконання роботи**

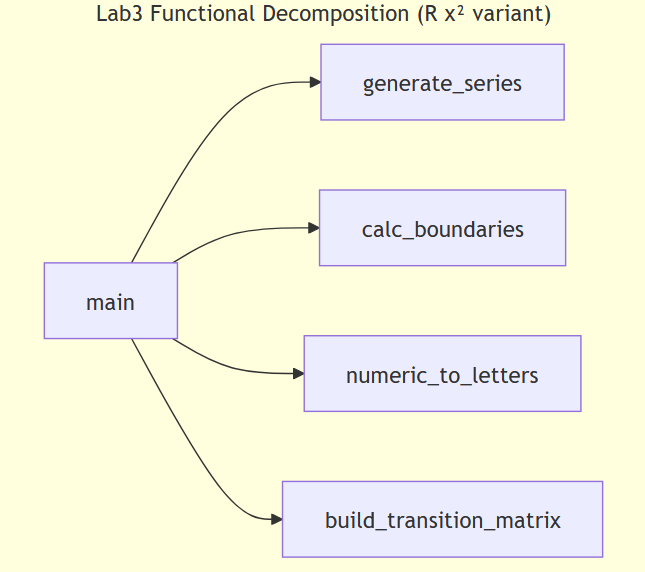
**Нижче наведено логічні блоки програми:**

1. Параметри розподілу
2. Генерація або зчитування числового ряду
3. Обчислення меж інтервалів через χ²-квантілі
4. Перетворення чисел у літери
5. Побудова матриці передування
6. Основна функція
7. Запуск

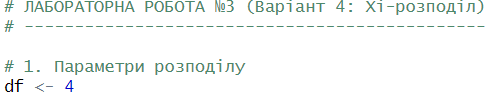
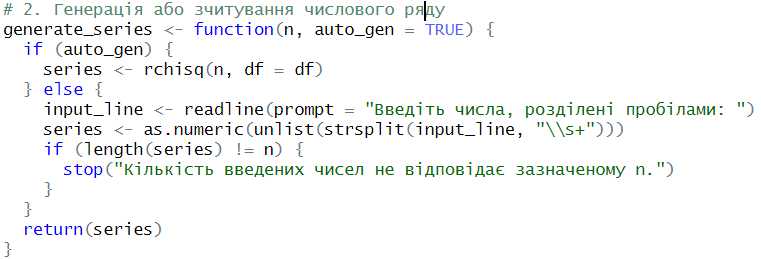
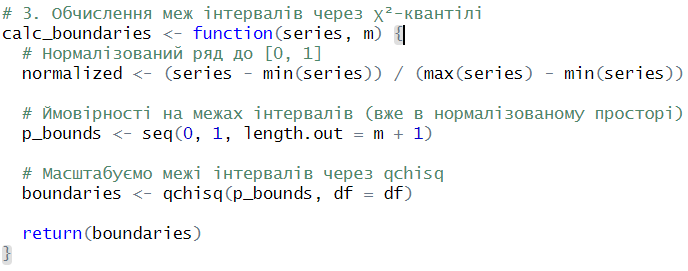
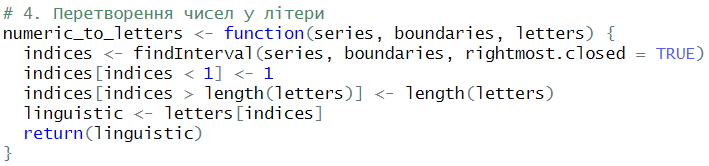
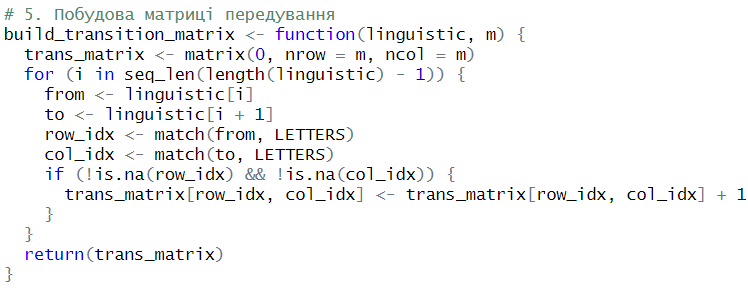
**Давайте розглянемо код для кожного пункту, в якому наведено коментарі українською мовою для кращого розуміння:**

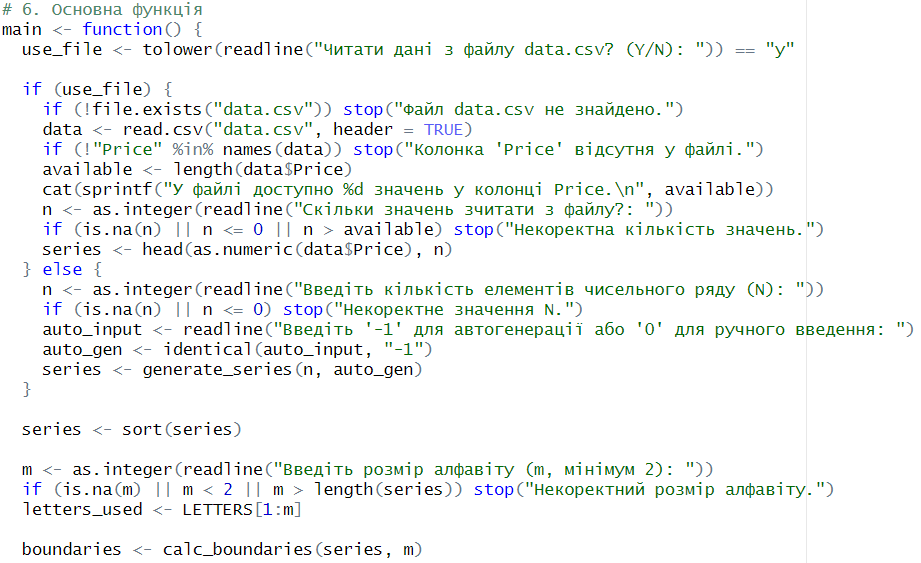
Інтерфейс RStudio for R Lang для запуску програми

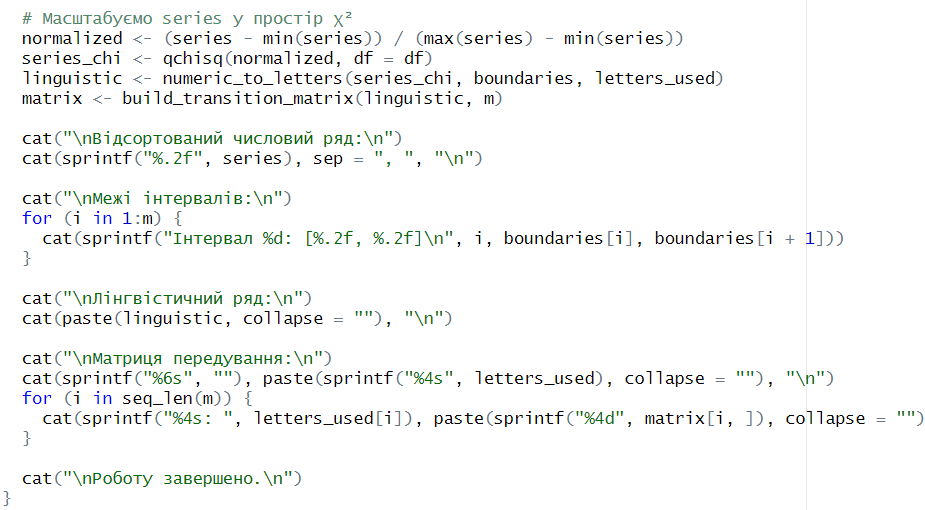


Функціональна декомпозиція і діаграма виконання програми:  


Етапи:

1. 
2. 
3. 
4. 
5. 

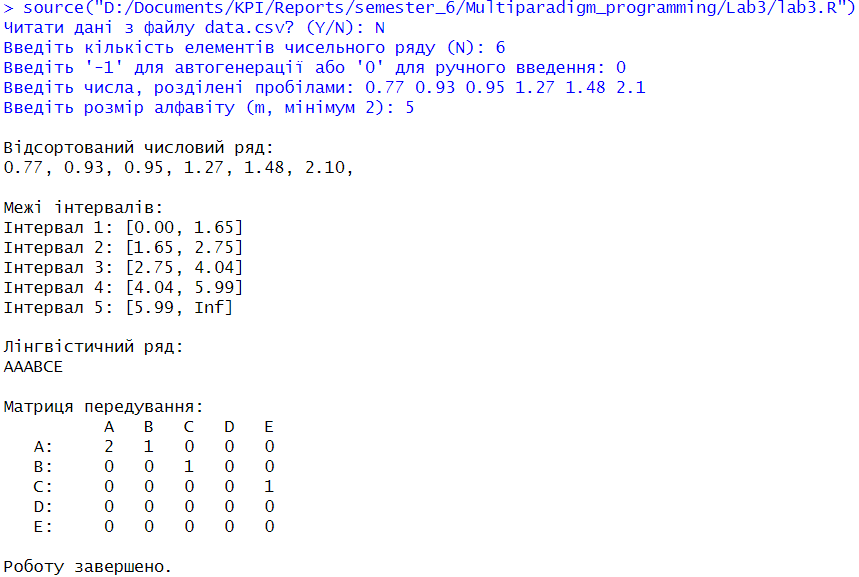




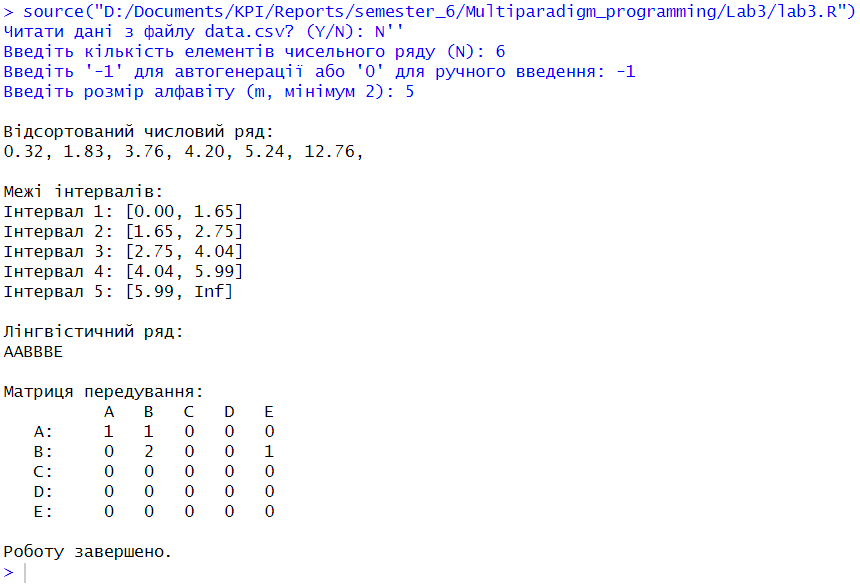
1. 

**Вхідні данні і результат виконання програми:**

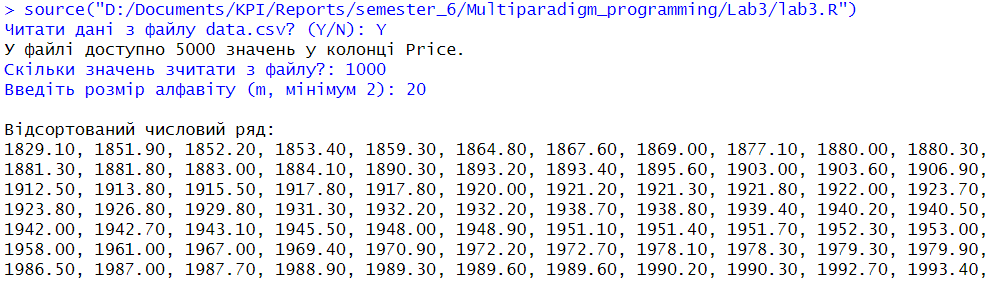
Введення значень від руки:

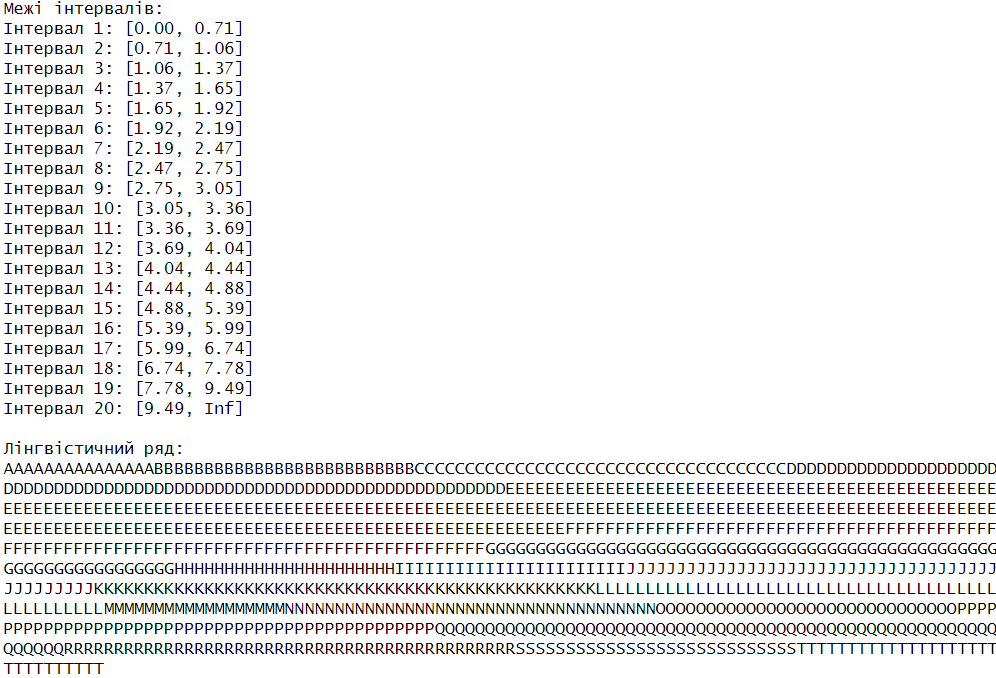


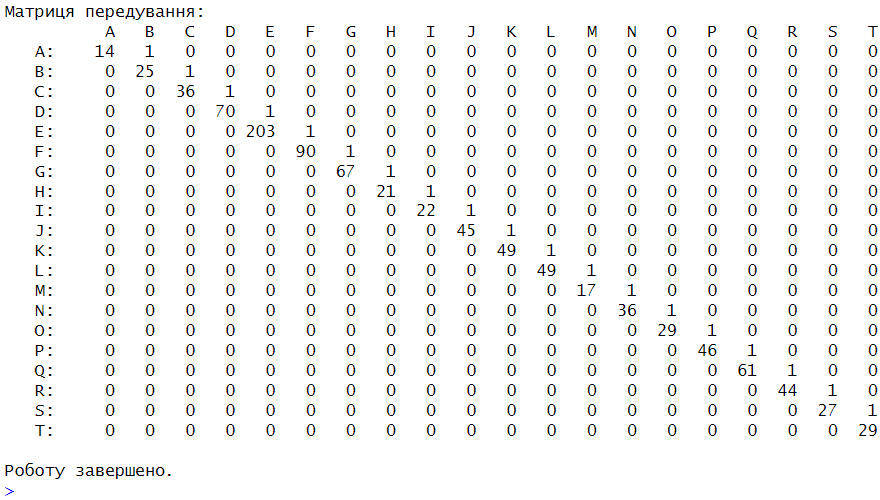
Генерація випадкових значень:



Читання з файлу часового ряду “B-C-D-E-S&P 500 Historical Data” з архіву, а саме поля Price:



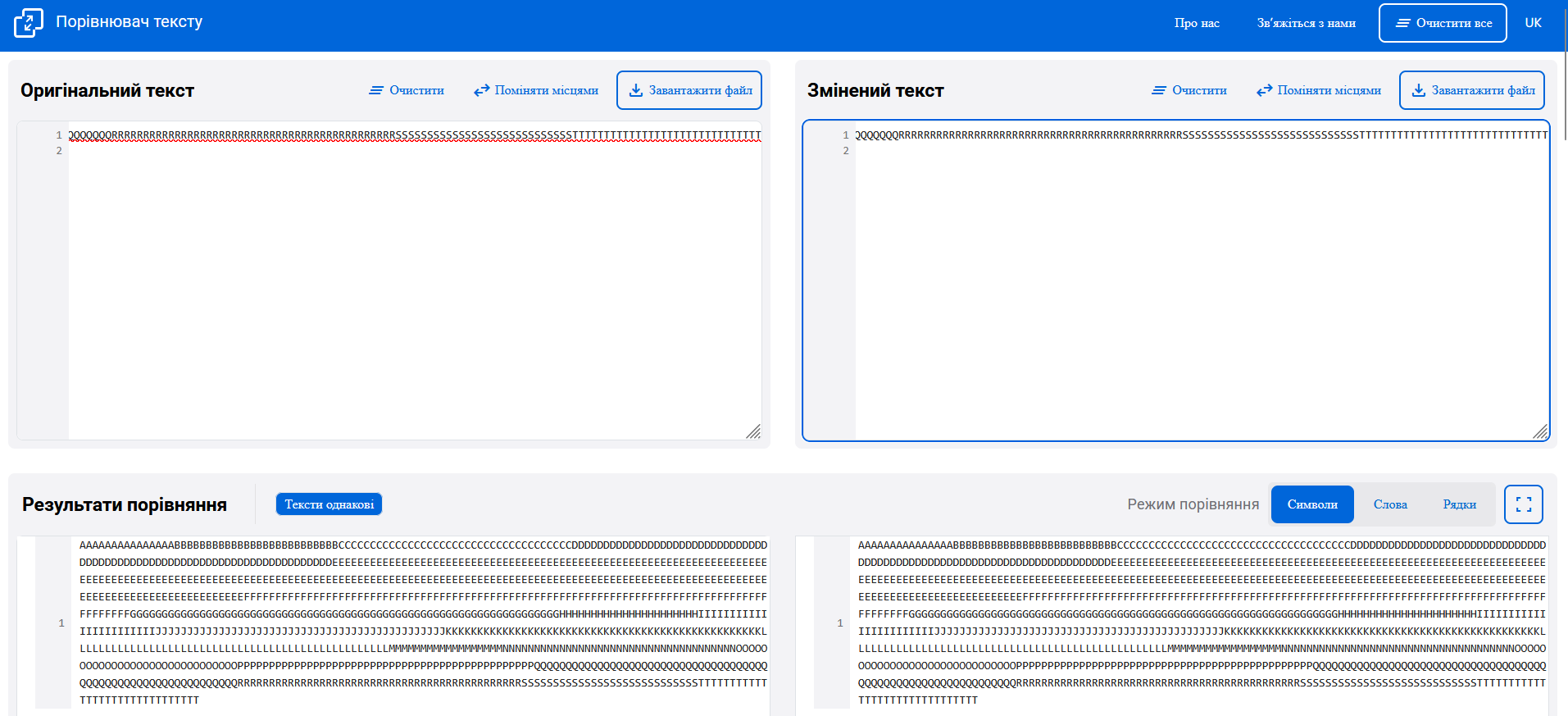




Поточний лінгвістичний ряд:

AAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLMMMMMMMMMMMMMMMMMMNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRSSSSSSSSSSSSSSSSSSSSSSSSSSSSTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

Порівняння лінгвістичних рядів лабораторної роботи 2 лабораторної роботи 3:

****

**Лінгвістичні ряди ідентичні.**

**Висновок**

У рамках лабораторної роботи №3 реалізовано програму на мові R, яка:

* Зчитує з консолі кількість елементів чисельного ряду (N), потужність алфавіту (m) та спосіб формування ряду (ручний ввід або автогенерація за Χі-розподілом з df = 4) ​
* Сортує вхідний числовий ряд у порядку зростання та визначає мінімальне і максимальне значення для подальшого розбиття на інтервали
* Обчислює межі m інтервалів за допомогою рівномірного поділу імовірності P[a,b] на інтервал [F(series\_min), F(series\_max)] і зворотної функції qchisq, тим самим гарантує, що кожен інтервал містить приблизно 1/m від загальної маси розподілу
* Перетворює відсортовані числа у лінгвістичний рядок, зіставляючи кожне значення з відповідною літерою алфавіту відповідно до його інтервалу
* Будує матрицю передування – підрахунок частот переходів між парами сусідніх символів у лінгвістичному рядку
* Виводить у консоль відсортований ряд, межі інтервалів, лінгвістичний рядок та матрицю передування у форматі, аналогічному до програм ЛР №1 та ЛР №2.

Результати тестів (ручний ввід і автогенерація) підтверджують правильність алгоритму:

* Відсортовані значення відповідають очікуваним порядкам.
* Інтервали побудовані так, щоб кожен містив рівну частку імовірності χ²-розподілу.
* Лінгвістичний рядок точно відображає відповідність чисел обраним буквам.
* Матриця передування коректно фіксує кількість появ кожної пари символів.

Таким чином, завдання ЛР №3 виконане повністю й відповідає вимогам лабораторної роботи з мультипарадигменного програмування.

**Додатки**

Код програми:

# ЛАБОРАТОРНА РОБОТА №3 (Варіант 4: Χі-розподіл)

# ----------------------------------------------

# 1. Параметри розподілу

df <- 4

# 2. Генерація або зчитування числового ряду

generate\_series <- function(n, auto\_gen = TRUE) {

if (auto\_gen) {

series <- rchisq(n, df = df)

} else {

input\_line <- readline(prompt = "Введіть числа, розділені пробілами: ")

series <- as.numeric(unlist(strsplit(input\_line, "\\s+")))

if (length(series) != n) {

stop("Кількість введених чисел не відповідає зазначеному n.")

}

}

return(series)

}

# 3. Обчислення меж інтервалів через χ²-квантілі

calc\_boundaries <- function(series, m) {

# Нормалізований ряд до [0, 1]

normalized <- (series - min(series)) / (max(series) - min(series))

# Ймовірності на межах інтервалів (вже в нормалізованому просторі)

p\_bounds <- seq(0, 1, length.out = m + 1)

# Масштабуємо межі інтервалів через qchisq

boundaries <- qchisq(p\_bounds, df = df)

return(boundaries)

}

# 4. Перетворення чисел у літери

numeric\_to\_letters <- function(series, boundaries, letters) {

indices <- findInterval(series, boundaries, rightmost.closed = TRUE)

indices[indices < 1] <- 1

indices[indices > length(letters)] <- length(letters)

linguistic <- letters[indices]

return(linguistic)

}

# 5. Побудова матриці передування

build\_transition\_matrix <- function(linguistic, m) {

trans\_matrix <- matrix(0, nrow = m, ncol = m)

for (i in seq\_len(length(linguistic) - 1)) {

from <- linguistic[i]

to <- linguistic[i + 1]

row\_idx <- match(from, LETTERS)

col\_idx <- match(to, LETTERS)

if (!is.na(row\_idx) && !is.na(col\_idx)) {

trans\_matrix[row\_idx, col\_idx] <- trans\_matrix[row\_idx, col\_idx] + 1

}

}

return(trans\_matrix)

}

# 6. Основна функція

main <- function() {

use\_file <- tolower(readline("Читати дані з файлу data.csv? (Y/N): ")) == "y"

if (use\_file) {

if (!file.exists("data.csv")) stop("Файл data.csv не знайдено.")

data <- read.csv("data.csv", header = TRUE)

if (!"Price" %in% names(data)) stop("Колонка 'Price' відсутня у файлі.")

available <- length(data$Price)

cat(sprintf("У файлі доступно %d значень у колонці Price.\n", available))

n <- as.integer(readline("Скільки значень зчитати з файлу?: "))

if (is.na(n) || n <= 0 || n > available) stop("Некоректна кількість значень.")

series <- head(as.numeric(data$Price), n)

} else {

n <- as.integer(readline("Введіть кількість елементів чисельного ряду (N): "))

if (is.na(n) || n <= 0) stop("Некоректне значення N.")

auto\_input <- readline("Введіть '-1' для автогенерації або '0' для ручного введення: ")

auto\_gen <- identical(auto\_input, "-1")

series <- generate\_series(n, auto\_gen)

}

series <- sort(series)

m <- as.integer(readline("Введіть розмір алфавіту (m, мінімум 2): "))

if (is.na(m) || m < 2 || m > length(series)) stop("Некоректний розмір алфавіту.")

letters\_used <- LETTERS[1:m]

boundaries <- calc\_boundaries(series, m)

# Масштабуємо series у простір χ²

normalized <- (series - min(series)) / (max(series) - min(series))

series\_chi <- qchisq(normalized, df = df)

linguistic <- numeric\_to\_letters(series\_chi, boundaries, letters\_used)

matrix <- build\_transition\_matrix(linguistic, m)

cat("\nВідсортований числовий ряд:\n")

cat(sprintf("%.2f", series), sep = ", ", "\n")

cat("\nМежі інтервалів:\n")

for (i in 1:m) {

cat(sprintf("Інтервал %d: [%.2f, %.2f]\n", i, boundaries[i], boundaries[i + 1]))

}

cat("\nЛінгвістичний ряд:\n")

cat(paste(linguistic, collapse = ""), "\n")

cat("\nМатриця передування:\n")

cat(sprintf("%6s", ""), paste(sprintf("%4s", letters\_used), collapse = ""), "\n")

for (i in seq\_len(m)) {

cat(sprintf("%4s: ", letters\_used[i]), paste(sprintf("%4d", matrix[i, ]), collapse = ""), "\n")

}

cat("\nРоботу завершено.\n")

}

# 7. Запуск

main()