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 №5**

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІС-23

Шимків Мирослав

Київ 2025

**ЛАБОРАТОРНАЯ РОБОТА №5**

**Завдання**: за допомогою продукційного програмування реалізувати перетворення чисельного ряду до лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів.

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

**Вихідні дані**: лінгвістичний ряд.

**Мова програмування**: CLIPS.

**Варіант 24**

(завдання 4 відповідно)

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

**Виконання роботи**

**Нижче наведено логічні блоки програми:**

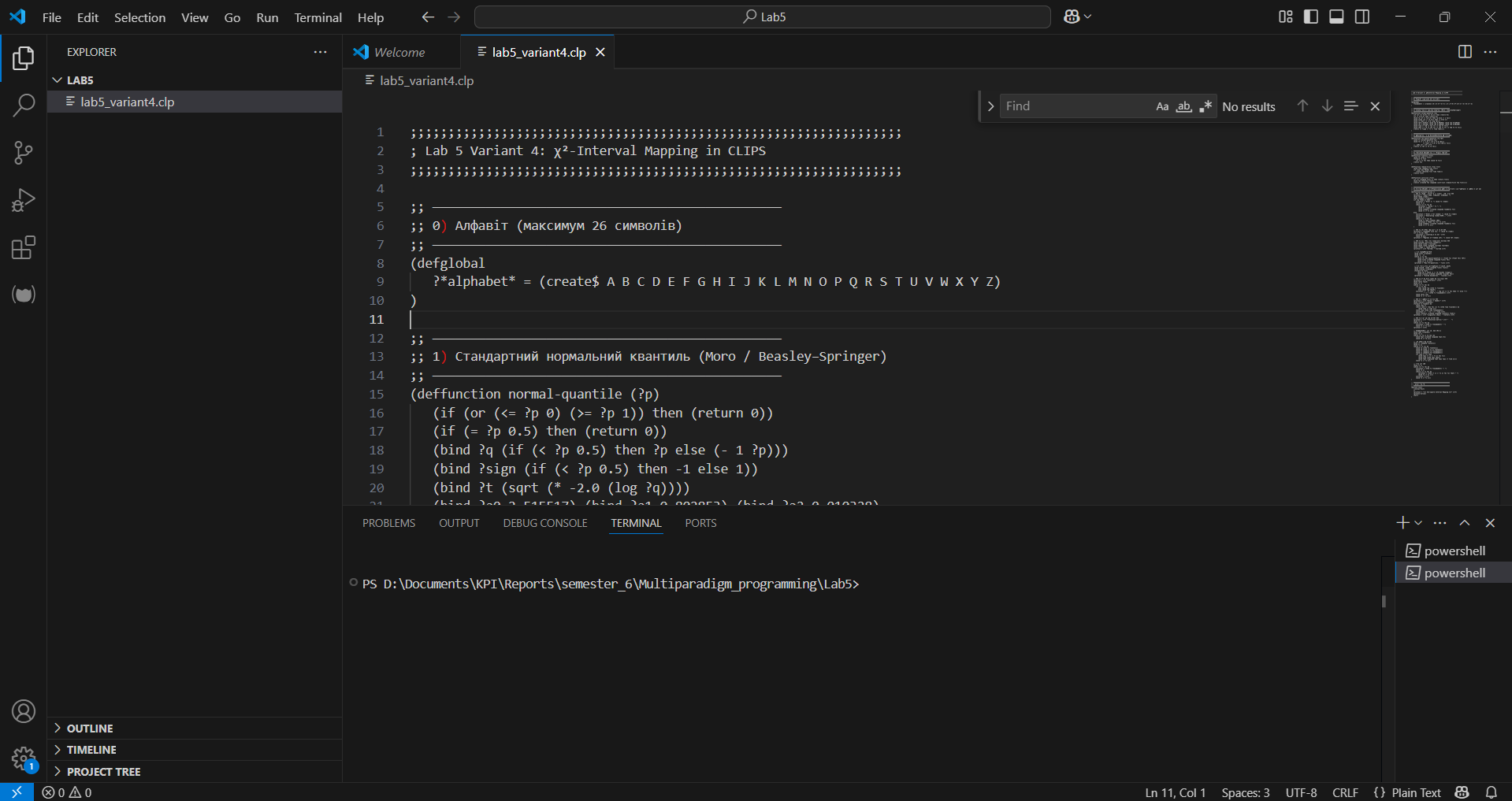
1. **Ініціалізація**  
   Створення глобального алфавіту (максимум 26 символів).
2. **Обчислення нормального квантіля**  
   Функція normal-quantile за алгоритмом Moro / Beasley–Springer.
3. **Обчислення χ²-квантіля**  
   Функція chi-square-quantile за апроксимацією Wilson–Hilferty.
4. **Сортування списку**  
   Допоміжні функції find-min, remove-first, sort-list.
5. **Обробка серії (process-series)**

Вибір режиму (ручне введення чи випадкове) та зчитування/генерація ряду чисел.

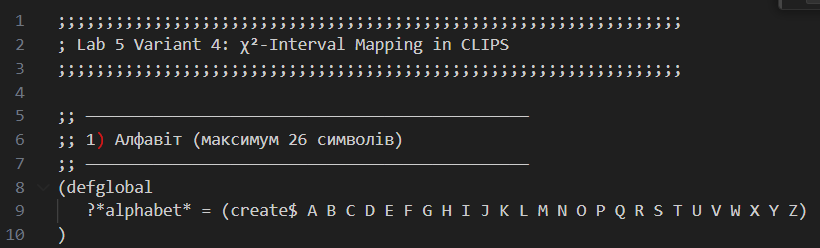
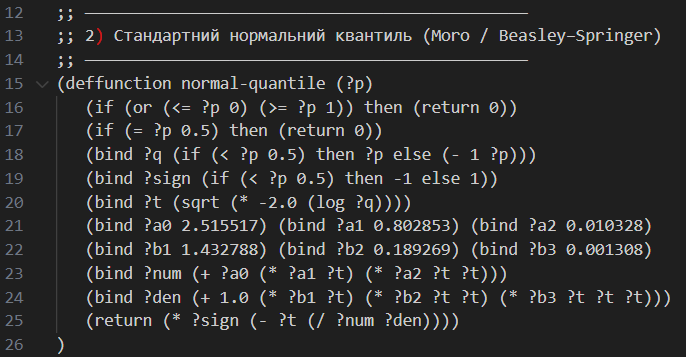
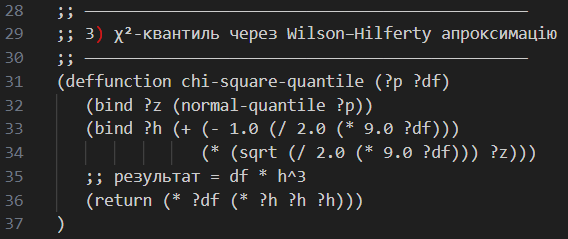
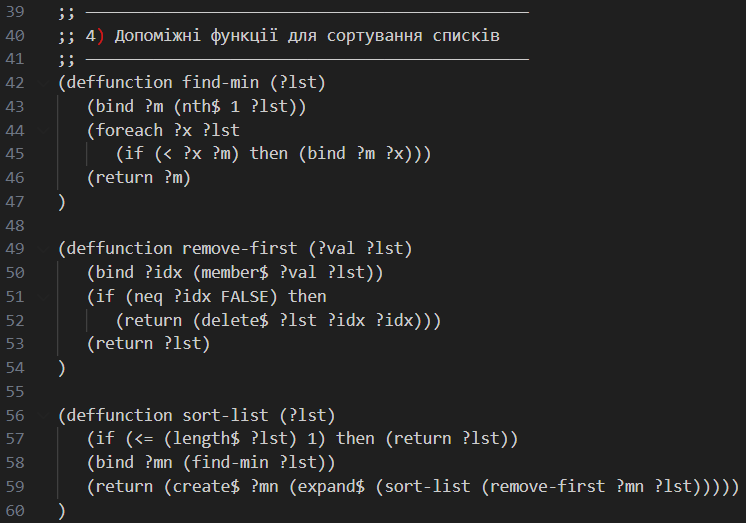
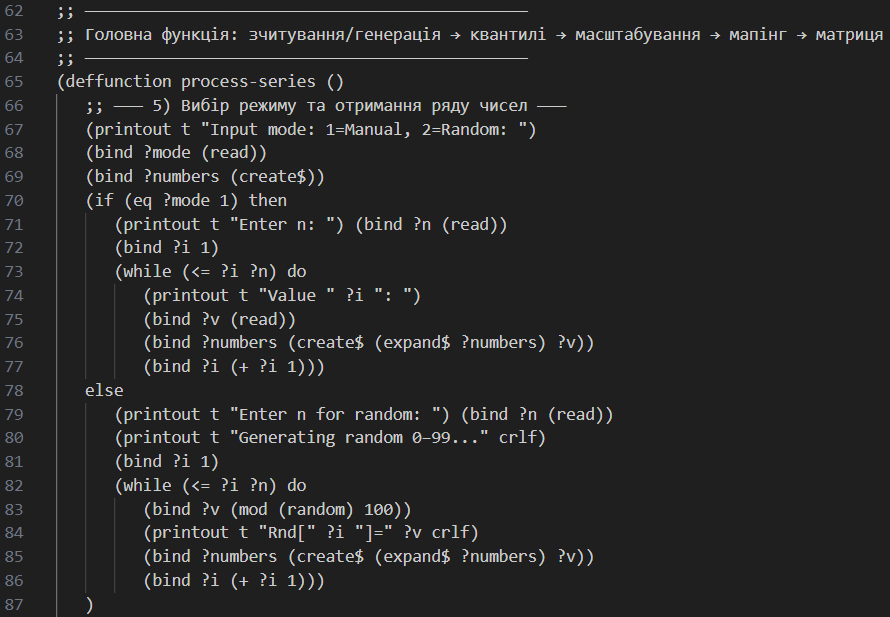
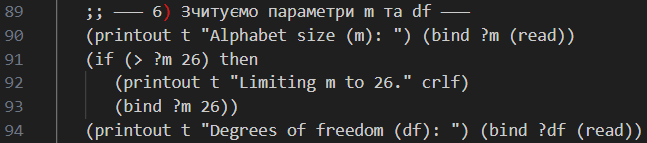
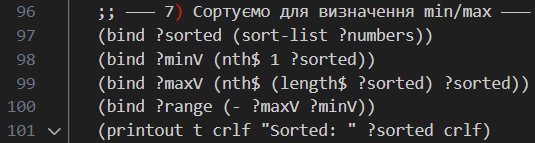
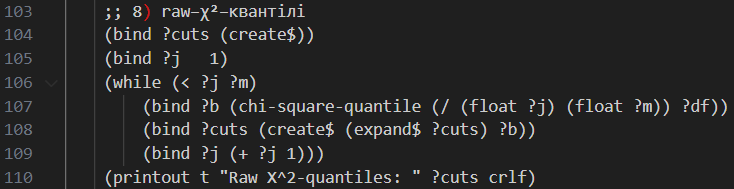
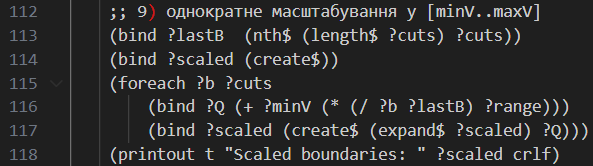
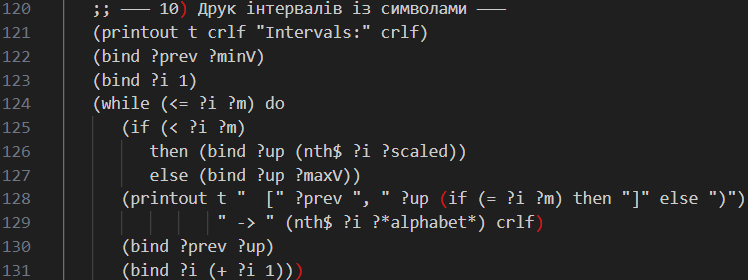
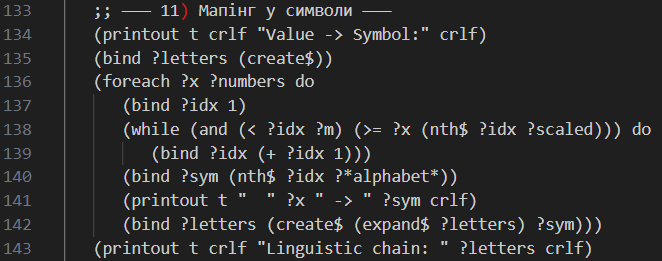
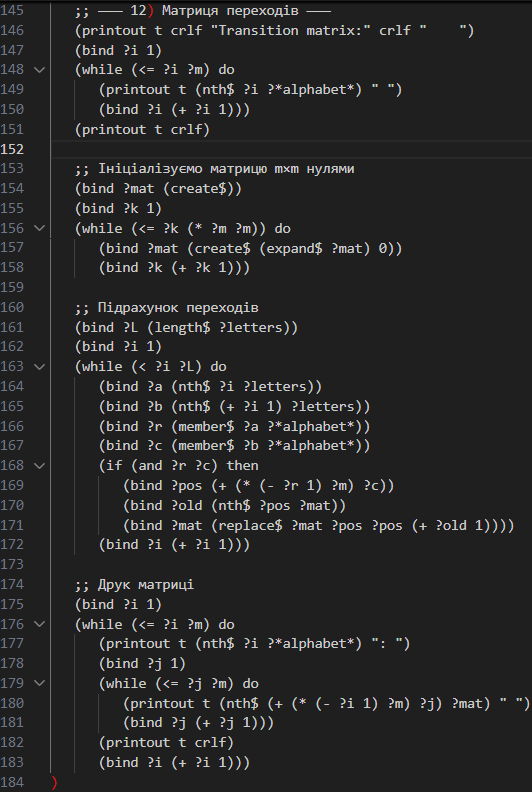
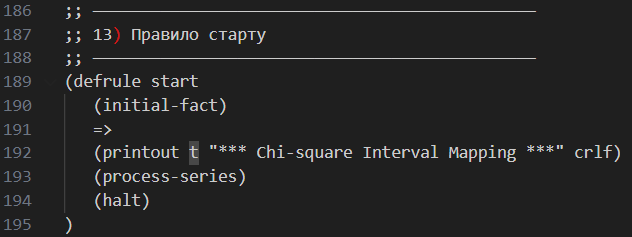
1. Зчитування параметрів: розмір алфавіту m і ступені свободи df.
2. Сортування отриманого ряду для визначення min/max.
3. Обчислення «Raw χ²-quantiles» (виправлено підпис).
4. Масштабування квантілів у діапазон [min..max].
5. Друк інтервалів із відповідними літерами алфавіту.
6. Мапінг кожного числа у символ — формування лінгвістичного ланцюжка.
7. Побудова та друк матриці переходів (частотна матриця m×m).
8. **Правило старту**  
   Дефініція правила start для автоматичного запуску process-series.

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

Інтерфейс Visual Studio Code for CLIPS для запуску програми

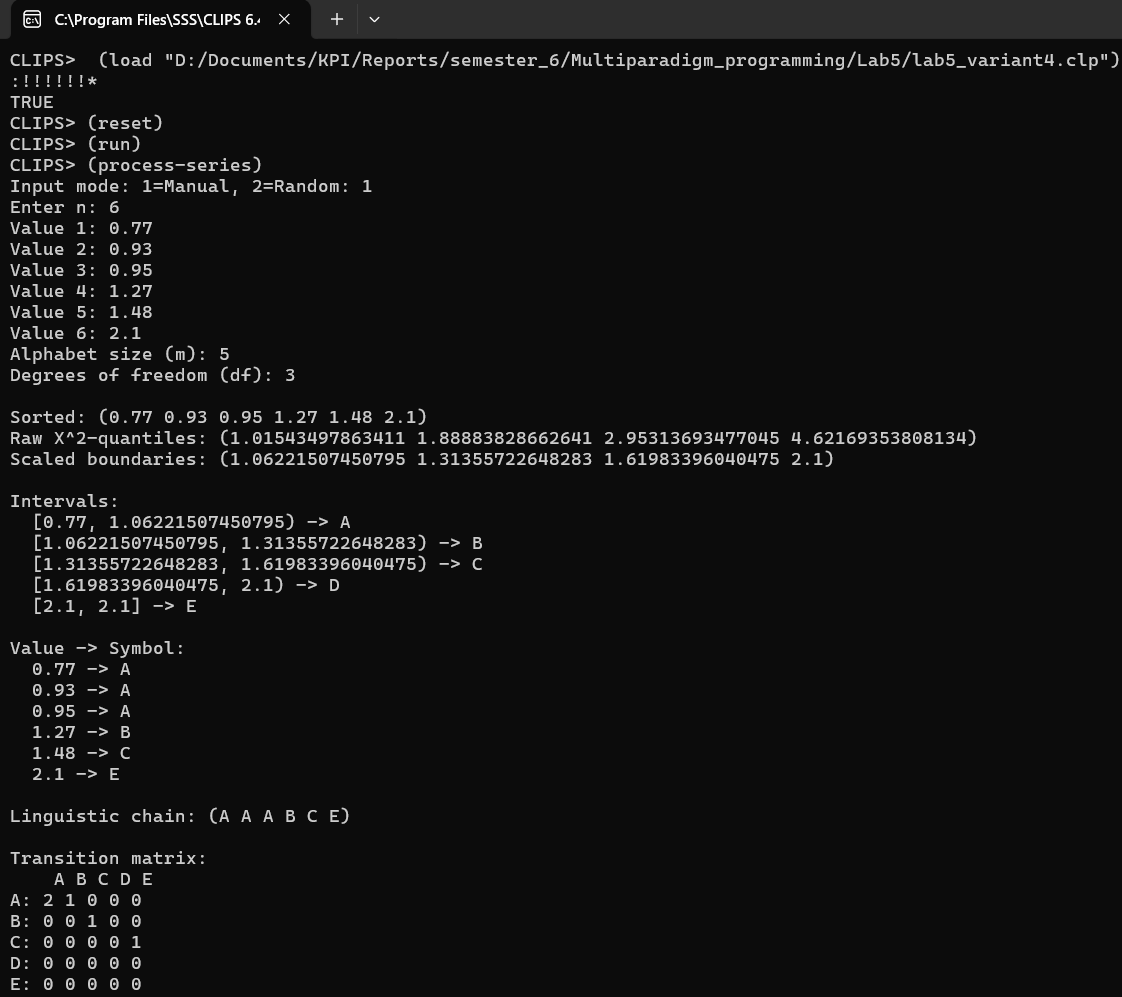


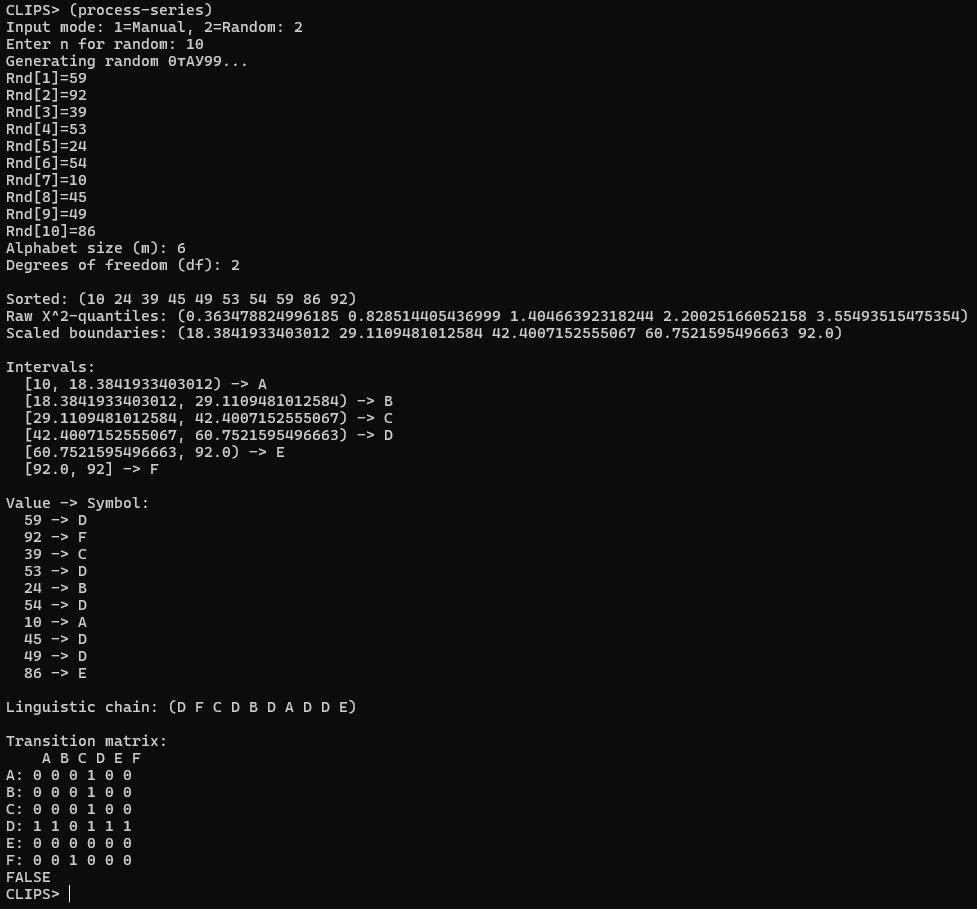
Етапи:

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 

**Вхідні данні і результат виконання програми:**

Введення значень від руки:



Генерація випадкових значень: 

**Висновок**

У рамках лабораторної роботи №5 реалізовано програму на мові CLIPS, яка:

* Зчитує режим введення (ручний або випадкова генерація), кількість елементів ряду (N), розмір алфавіту (m) та ступені свободи (df) для χ²-розподілу.
* Сортує ряд у порядку зростання та визначає мінімальне й максимальне значення.
* Обчислює «raw» χ²-квантілі, масштабує їх у відрізок [min, max] і виводить обидва набори меж.
* Мапує кожне значення в символ алфавіту за інтервалом, формуючи лінгвістичний рядок.
* Будує матрицю частот переходів між сусідніми символами.
* Виводить у консоль відсортований ряд, межі інтервалів, лінгвістичний рядок та матрицю переходів.

Результати тестів (ручне введення й автогенерація) доводять коректність алгоритму:

* Відсортовані значення відповідають очікуваним порядкам.
* Межі інтервалів забезпечують приблизно рівномірний розподіл маси χ²-розподілу.
* Лінгвістичний рядок точно відображає відповідність чисел літерам.
* Матриця переходів правильно фіксує частоти появ кожної пари символів.

Отже, завдання ЛР №5 виконане повністю й відповідає вимогам лабораторної роботи з продукційного програмування.

**Додатки**

Код програми:

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; Lab 5 Variant 4: χ²‑Interval Mapping in CLIPS

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; ——————————————————————————————————————————————

;; 1) Алфавіт (максимум 26 символів)

;; ——————————————————————————————————————————————

(defglobal

?\*alphabet\* = (create$ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)

)

;; ——————————————————————————————————————————————

;; 2) Стандартний нормальний квантиль (Moro / Beasley–Springer)

;; ——————————————————————————————————————————————

(deffunction normal-quantile (?p)

(if (or (<= ?p 0) (>= ?p 1)) then (return 0))

(if (= ?p 0.5) then (return 0))

(bind ?q (if (< ?p 0.5) then ?p else (- 1 ?p)))

(bind ?sign (if (< ?p 0.5) then -1 else 1))

(bind ?t (sqrt (\* -2.0 (log ?q))))

(bind ?a0 2.515517) (bind ?a1 0.802853) (bind ?a2 0.010328)

(bind ?b1 1.432788) (bind ?b2 0.189269) (bind ?b3 0.001308)

(bind ?num (+ ?a0 (\* ?a1 ?t) (\* ?a2 ?t ?t)))

(bind ?den (+ 1.0 (\* ?b1 ?t) (\* ?b2 ?t ?t) (\* ?b3 ?t ?t ?t)))

(return (\* ?sign (- ?t (/ ?num ?den))))

)

;; ——————————————————————————————————————————————

;; 3) χ²‑квантиль через Wilson–Hilferty апроксимацію

;; ——————————————————————————————————————————————

(deffunction chi-square-quantile (?p ?df)

(bind ?z (normal-quantile ?p))

(bind ?h (+ (- 1.0 (/ 2.0 (\* 9.0 ?df)))

(\* (sqrt (/ 2.0 (\* 9.0 ?df))) ?z)))

;; результат = df \* h^3

(return (\* ?df (\* ?h ?h ?h)))

)

;; ——————————————————————————————————————————————

;; 4) Допоміжні функції для сортування списків

;; ——————————————————————————————————————————————

(deffunction find-min (?lst)

(bind ?m (nth$ 1 ?lst))

(foreach ?x ?lst

(if (< ?x ?m) then (bind ?m ?x)))

(return ?m)

)

(deffunction remove-first (?val ?lst)

(bind ?idx (member$ ?val ?lst))

(if (neq ?idx FALSE) then

(return (delete$ ?lst ?idx ?idx)))

(return ?lst)

)

(deffunction sort-list (?lst)

(if (<= (length$ ?lst) 1) then (return ?lst))

(bind ?mn (find-min ?lst))

(return (create$ ?mn (expand$ (sort-list (remove-first ?mn ?lst)))))

)

;; ——————————————————————————————————————————————

;; Головна функція: зчитування/генерація → квантилі → масштабування → мапінг → матриця

;; ——————————————————————————————————————————————

(deffunction process-series ()

;; ——— 5) Вибір режиму та отримання ряду чисел ———

(printout t "Input mode: 1=Manual, 2=Random: ")

(bind ?mode (read))

(bind ?numbers (create$))

(if (eq ?mode 1) then

(printout t "Enter n: ") (bind ?n (read))

(bind ?i 1)

(while (<= ?i ?n) do

(printout t "Value " ?i ": ")

(bind ?v (read))

(bind ?numbers (create$ (expand$ ?numbers) ?v))

(bind ?i (+ ?i 1)))

else

(printout t "Enter n for random: ") (bind ?n (read))

(printout t "Generating random 0–99..." crlf)

(bind ?i 1)

(while (<= ?i ?n) do

(bind ?v (mod (random) 100))

(printout t "Rnd[" ?i "]=" ?v crlf)

(bind ?numbers (create$ (expand$ ?numbers) ?v))

(bind ?i (+ ?i 1)))

)

;; ——— 6) Зчитуємо параметри m та df ———

(printout t "Alphabet size (m): ") (bind ?m (read))

(if (> ?m 26) then

(printout t "Limiting m to 26." crlf)

(bind ?m 26))

(printout t "Degrees of freedom (df): ") (bind ?df (read))

;; ——— 7) Сортуємо для визначення min/max ———

(bind ?sorted (sort-list ?numbers))

(bind ?minV (nth$ 1 ?sorted))

(bind ?maxV (nth$ (length$ ?sorted) ?sorted))

(bind ?range (- ?maxV ?minV))

(printout t crlf "Sorted: " ?sorted crlf)

;; 8) raw–χ²−квантілі

(bind ?cuts (create$))

(bind ?j 1)

(while (< ?j ?m)

(bind ?b (chi-square-quantile (/ (float ?j) (float ?m)) ?df))

(bind ?cuts (create$ (expand$ ?cuts) ?b))

(bind ?j (+ ?j 1)))

(printout t "Raw X^2-quantiles: " ?cuts crlf)

;; 9) однократне масштабування у [minV..maxV]

(bind ?lastB (nth$ (length$ ?cuts) ?cuts))

(bind ?scaled (create$))

(foreach ?b ?cuts

(bind ?Q (+ ?minV (\* (/ ?b ?lastB) ?range)))

(bind ?scaled (create$ (expand$ ?scaled) ?Q)))

(printout t "Scaled boundaries: " ?scaled crlf)

;; ——— 10) Друк інтервалів із символами ———

(printout t crlf "Intervals:" crlf)

(bind ?prev ?minV)

(bind ?i 1)

(while (<= ?i ?m) do

(if (< ?i ?m)

then (bind ?up (nth$ ?i ?scaled))

else (bind ?up ?maxV))

(printout t " [" ?prev ", " ?up (if (= ?i ?m) then "]" else ")")

" -> " (nth$ ?i ?\*alphabet\*) crlf)

(bind ?prev ?up)

(bind ?i (+ ?i 1)))

;; ——— 11) Мапінг у символи ———

(printout t crlf "Value -> Symbol:" crlf)

(bind ?letters (create$))

(foreach ?x ?numbers do

(bind ?idx 1)

(while (and (< ?idx ?m) (>= ?x (nth$ ?idx ?scaled))) do

(bind ?idx (+ ?idx 1)))

(bind ?sym (nth$ ?idx ?\*alphabet\*))

(printout t " " ?x " -> " ?sym crlf)

(bind ?letters (create$ (expand$ ?letters) ?sym)))

(printout t crlf "Linguistic chain: " ?letters crlf)

;; ——— 12) Матриця переходів ———

(printout t crlf "Transition matrix:" crlf " ")

(bind ?i 1)

(while (<= ?i ?m) do

(printout t (nth$ ?i ?\*alphabet\*) " ")

(bind ?i (+ ?i 1)))

(printout t crlf)

;; Ініціалізуємо матрицю m×m нулями

(bind ?mat (create$))

(bind ?k 1)

(while (<= ?k (\* ?m ?m)) do

(bind ?mat (create$ (expand$ ?mat) 0))

(bind ?k (+ ?k 1)))

;; Підрахунок переходів

(bind ?L (length$ ?letters))

(bind ?i 1)

(while (< ?i ?L) do

(bind ?a (nth$ ?i ?letters))

(bind ?b (nth$ (+ ?i 1) ?letters))

(bind ?r (member$ ?a ?\*alphabet\*))

(bind ?c (member$ ?b ?\*alphabet\*))

(if (and ?r ?c) then

(bind ?pos (+ (\* (- ?r 1) ?m) ?c))

(bind ?old (nth$ ?pos ?mat))

(bind ?mat (replace$ ?mat ?pos ?pos (+ ?old 1))))

(bind ?i (+ ?i 1)))

;; Друк матриці

(bind ?i 1)

(while (<= ?i ?m) do

(printout t (nth$ ?i ?\*alphabet\*) ": ")

(bind ?j 1)

(while (<= ?j ?m) do

(printout t (nth$ (+ (\* (- ?i 1) ?m) ?j) ?mat) " ")

(bind ?j (+ ?j 1)))

(printout t crlf)

(bind ?i (+ ?i 1)))

)

;; ——————————————————————————————————————————————

;; 13) Правило старту

;; ——————————————————————————————————————————————

(defrule start

(initial-fact)

=>

(printout t "\*\*\* Chi-square Interval Mapping \*\*\*" crlf)

(process-series)

(halt)

)