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 №4**

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІС-23

Шимків Мирослав

Київ 2025

**ЛАБОРАТОРНАЯ РОБОТА №4**

**Завдання**: за допомогою логічного 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**: Prolog і його клони.

**Варіант 24**

(завдання 4 відповідно)

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

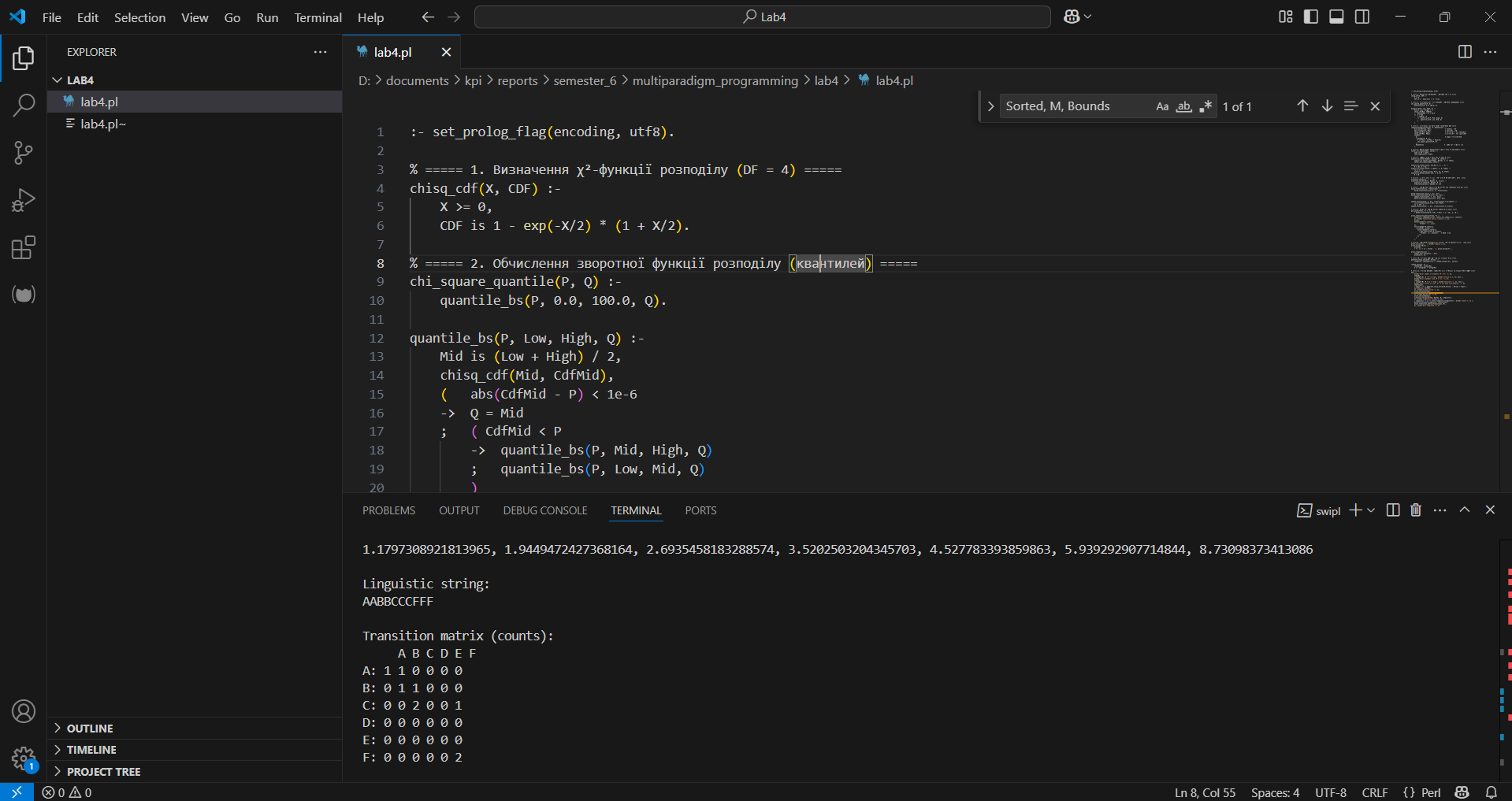
**Виконання роботи**

**Нижче наведено логічні блоки програми:**

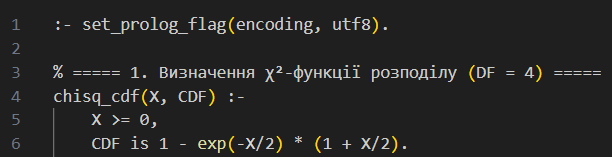
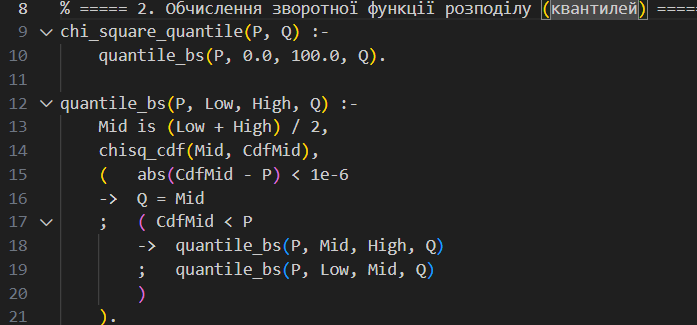
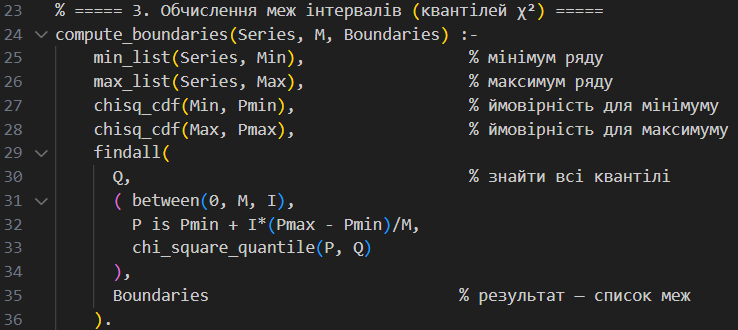
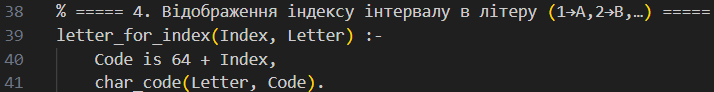
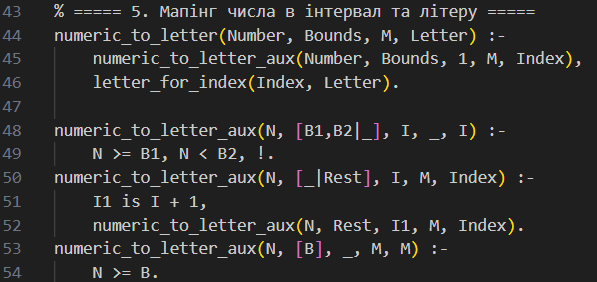
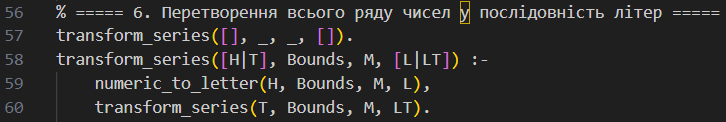
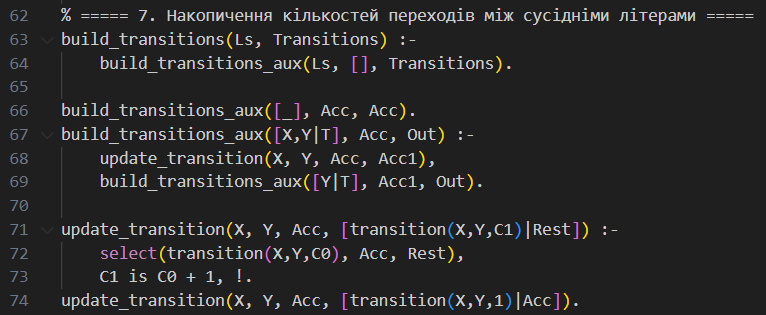
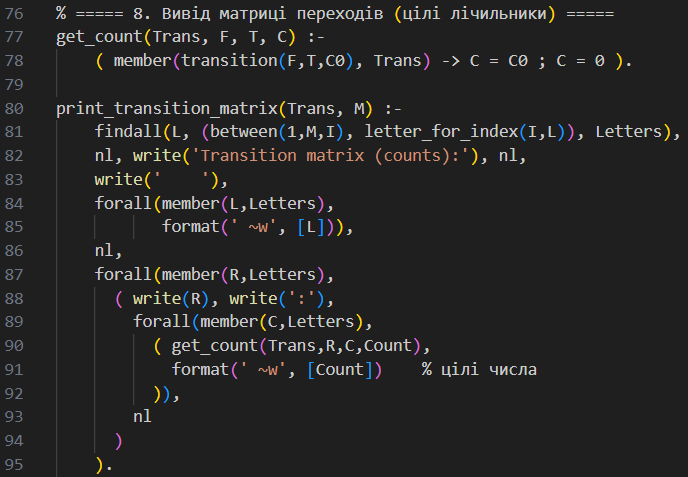
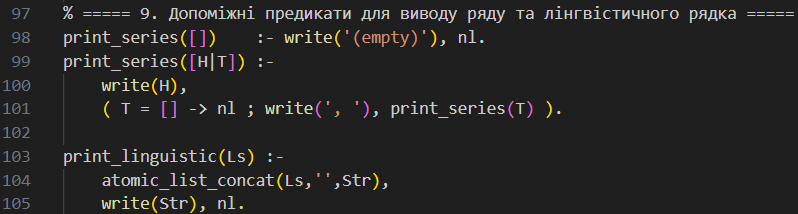
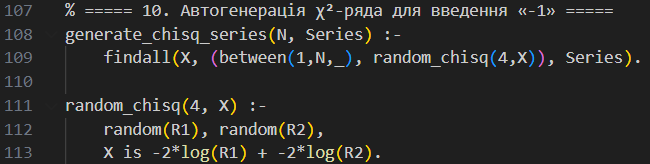
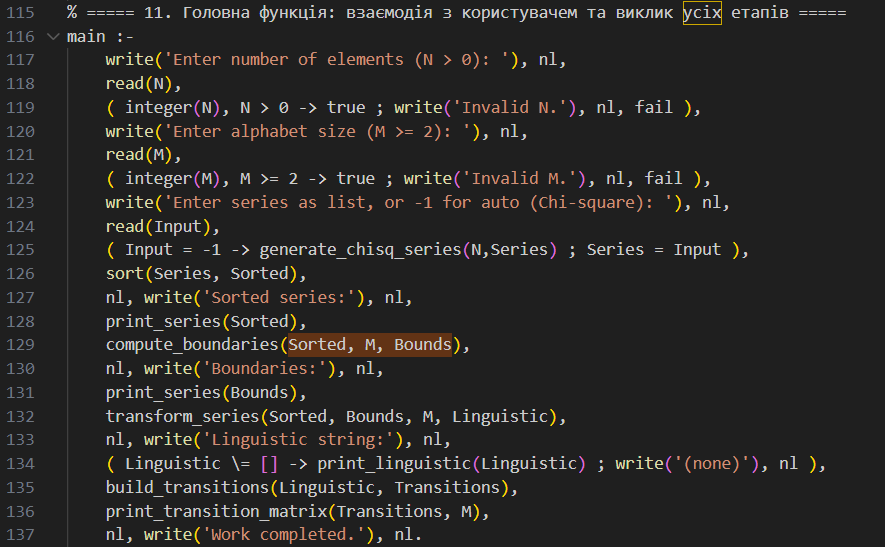
1. Визначення χ²-функції розподілу.
2. Обчислення зворотної функції розподілу (квантилей).
3. Обчислення меж інтервалів.
4. Відображення індексу інтервалу в літеру.
5. Мапінг числа в інтервал та літеру.
6. Перетворення всього ряду чисел у послідовність літер.
7. Накопичення кількостей переходів між сусідніми літерами.
8. Вивід матриці переходів (цілі лічильники).
9. Допоміжні предикати для виводу числового ряду та лінгвістичного рядка.
10. Автогенерація χ²-ряда для введення «-1».
11. Головна функція main/0: послідовне виклику всіх етапів та взаємодія з користувачем;

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

Інтерфейс Visual Studio Code for Prolog для запуску програми

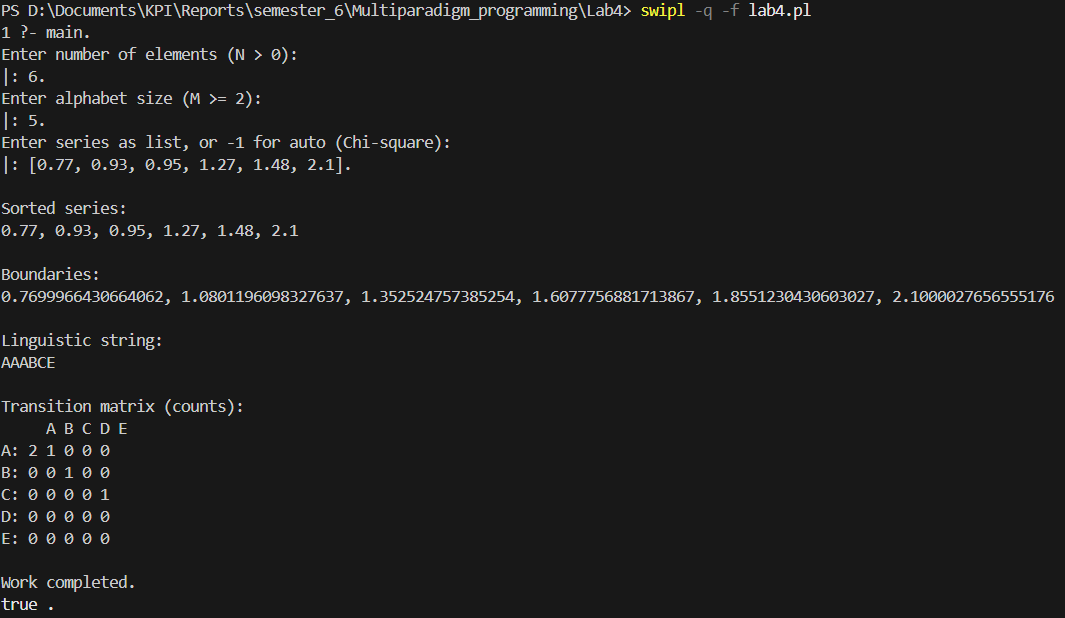


Етапи:

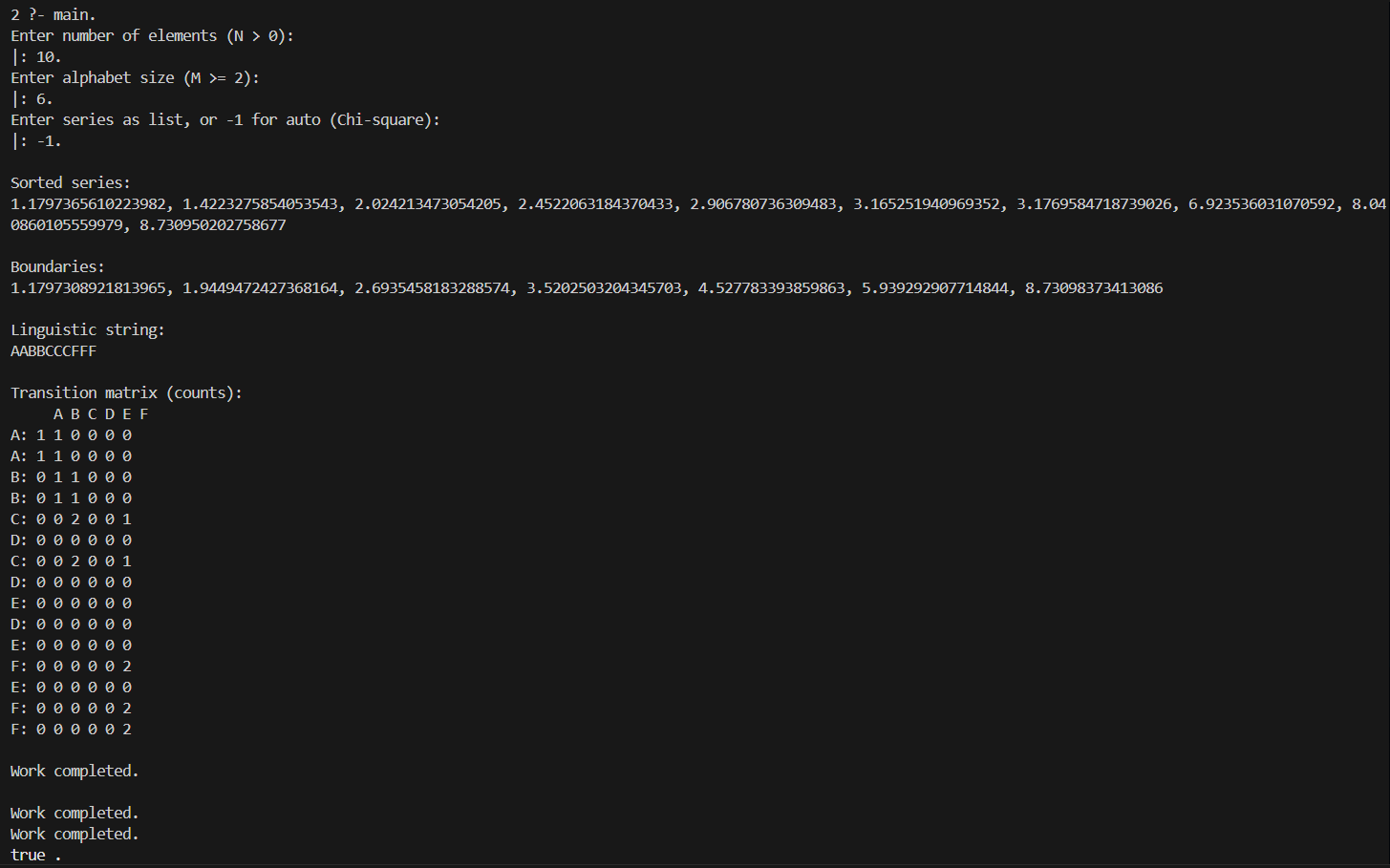
1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 

**Вхідні данні і результат виконання програми:**

Введення значень від руки:



Генерація випадкових значень:



**Висновок**

У рамках лабораторної роботи №4 реалізовано програму на мові Prolog, яка:

* Зчитує з консолі кількість елементів числового ряду (N > 0), розмір алфавіту (M ≥ 2) та сам ряд (ручний ввід або автогенерація χ²-послідовності з df = 4).
* Сортує вхідний числовий ряд у порядку зростання та обчислює мінімальне й максимальне значення.
* Обчислює M + 1 меж χ²-квантілей між Min і Max за допомогою оберненої кумулятивної функції та бінарного пошуку.
* Перетворює відсортовані числа у лінгвістичний рядок, зіставляючи кожне значення з відповідною літерою алфавіту за індексом інтервалу.
* Будує матрицю передування — підрахунок частот переходів між кожною парою сусідніх символів.
* Виводить у консоль відсортований ряд, межі інтервалів, лінгвістичний рядок та матрицю передування із цілими значеннями частот.

Результати тестів із ручним введенням і автогенерацією підтверджують коректність алгоритму:

* Відсортовані значення відповідають очікуваним порядкам.
* Межі інтервалів обчислені через рівномірний розподіл ймовірності χ² і гарантують приблизно рівномірну місткість інтервалів.
* Лінгвістичний рядок точно відображає відповідність чисел обраним літерам.
* Матриця передування правильно фіксує кількість появ кожної пари символів.

Таким чином, завдання ЛР №4 виконане повністю й відповідає вимогам лабораторної роботи з логічного програмування.

**Додатки**

Код програми:

:- set\_prolog\_flag(encoding, utf8).

% ===== 1. Визначення χ²-функції розподілу (DF = 4) =====

chisq\_cdf(X, CDF) :-

    X >= 0,

    CDF is 1 - exp(-X/2) \* (1 + X/2).

% ===== 2. Обчислення зворотної функції розподілу (квантилей) =====

chi\_square\_quantile(P, Q) :-

    quantile\_bs(P, 0.0, 100.0, Q).

quantile\_bs(P, Low, High, Q) :-

    Mid is (Low + High) / 2,

    chisq\_cdf(Mid, CdfMid),

    (   abs(CdfMid - P) < 1e-6

    ->  Q = Mid

    ;   ( CdfMid < P

        ->  quantile\_bs(P, Mid, High, Q)

        ;   quantile\_bs(P, Low, Mid, Q)

        )

    ).

% ===== 3. Обчислення меж інтервалів (квантілей χ²) =====

compute\_boundaries(Series, M, Boundaries) :-

    min\_list(Series, Min),                 % мінімум ряду

    max\_list(Series, Max),                 % максимум ряду

    chisq\_cdf(Min, Pmin),                  % ймовірність для мінімуму

    chisq\_cdf(Max, Pmax),                  % ймовірність для максимуму

    findall(

      Q,                                   % знайти всі квантілі

      ( between(0, M, I),

        P is Pmin + I\*(Pmax - Pmin)/M,

        chi\_square\_quantile(P, Q)

      ),

      Boundaries                          % результат — список меж

    ).

% ===== 4. Відображення індексу інтервалу в літеру (1→A,2→B,…) =====

letter\_for\_index(Index, Letter) :-

    Code is 64 + Index,

    char\_code(Letter, Code).

% ===== 5. Мапінг числа в інтервал та літеру =====

numeric\_to\_letter(Number, Bounds, M, Letter) :-

    numeric\_to\_letter\_aux(Number, Bounds, 1, M, Index),

    letter\_for\_index(Index, Letter).

numeric\_to\_letter\_aux(N, [B1,B2|\_], I, \_, I) :-

    N >= B1, N < B2, !.

numeric\_to\_letter\_aux(N, [\_|Rest], I, M, Index) :-

    I1 is I + 1,

    numeric\_to\_letter\_aux(N, Rest, I1, M, Index).

numeric\_to\_letter\_aux(N, [B], \_, M, M) :-

    N >= B.

% ===== 6. Перетворення всього ряду чисел у послідовність літер =====

transform\_series([], \_, \_, []).

transform\_series([H|T], Bounds, M, [L|LT]) :-

    numeric\_to\_letter(H, Bounds, M, L),

    transform\_series(T, Bounds, M, LT).

% ===== 7. Накопичення кількостей переходів між сусідніми літерами =====

build\_transitions(Ls, Transitions) :-

    build\_transitions\_aux(Ls, [], Transitions).

build\_transitions\_aux([\_], Acc, Acc).

build\_transitions\_aux([X,Y|T], Acc, Out) :-

    update\_transition(X, Y, Acc, Acc1),

    build\_transitions\_aux([Y|T], Acc1, Out).

update\_transition(X, Y, Acc, [transition(X,Y,C1)|Rest]) :-

    select(transition(X,Y,C0), Acc, Rest),

    C1 is C0 + 1, !.

update\_transition(X, Y, Acc, [transition(X,Y,1)|Acc]).

% ===== 8. Вивід матриці переходів (цілі лічильники) =====

get\_count(Trans, F, T, C) :-

    ( member(transition(F,T,C0), Trans) -> C = C0 ; C = 0 ).

print\_transition\_matrix(Trans, M) :-

    findall(L, (between(1,M,I), letter\_for\_index(I,L)), Letters),

    nl, write('Transition matrix (counts):'), nl,

    write('    '),

    forall(member(L,Letters),

           format(' ~w', [L])),

    nl,

    forall(member(R,Letters),

      ( write(R), write(':'),

        forall(member(C,Letters),

          ( get\_count(Trans,R,C,Count),

            format(' ~w', [Count])    % цілі числа

          )),

        nl

      )

    ).

% ===== 9. Допоміжні предикати для виводу ряду та лінгвістичного рядка =====

print\_series([])    :- write('(empty)'), nl.

print\_series([H|T]) :-

    write(H),

    ( T = [] -> nl ; write(', '), print\_series(T) ).

print\_linguistic(Ls) :-

    atomic\_list\_concat(Ls,'',Str),

    write(Str), nl.

% ===== 10. Автогенерація χ²-ряда для введення «-1» =====

generate\_chisq\_series(N, Series) :-

    findall(X, (between(1,N,\_), random\_chisq(4,X)), Series).

random\_chisq(4, X) :-

    random(R1), random(R2),

    X is -2\*log(R1) + -2\*log(R2).

% ===== 11. Головна функція: взаємодія з користувачем та виклик усіх етапів =====

main :-

    write('Enter number of elements (N > 0): '), nl,

    read(N),

    ( integer(N), N > 0 -> true ; write('Invalid N.'), nl, fail ),

    write('Enter alphabet size (M >= 2): '), nl,

    read(M),

    ( integer(M), M >= 2 -> true ; write('Invalid M.'), nl, fail ),

    write('Enter series as list, or -1 for auto (Chi-square): '), nl,

    read(Input),

    ( Input = -1 -> generate\_chisq\_series(N,Series) ; Series = Input ),

    sort(Series, Sorted),

    nl, write('Sorted series:'), nl,

    print\_series(Sorted),

    compute\_boundaries(Sorted, M, Bounds),

    nl, write('Boundaries:'), nl,

    print\_series(Bounds),

    transform\_series(Sorted, Bounds, M, Linguistic),

    nl, write('Linguistic string:'), nl,

    ( Linguistic \= [] -> print\_linguistic(Linguistic) ; write('(none)'), nl ),

    build\_transitions(Linguistic, Transitions),

    print\_transition\_matrix(Transitions, M),

    nl, write('Work completed.'), nl.