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 №6**

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІС-23

Шимків Мирослав

Київ 2025

**ЛАБОРАТОРНАЯ РОБОТА №6**

**Завдання**: за допомогою системи алгебраїчного 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**: MAXIMA.

**Варіант 24**

(завдання 4 відповідно)

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

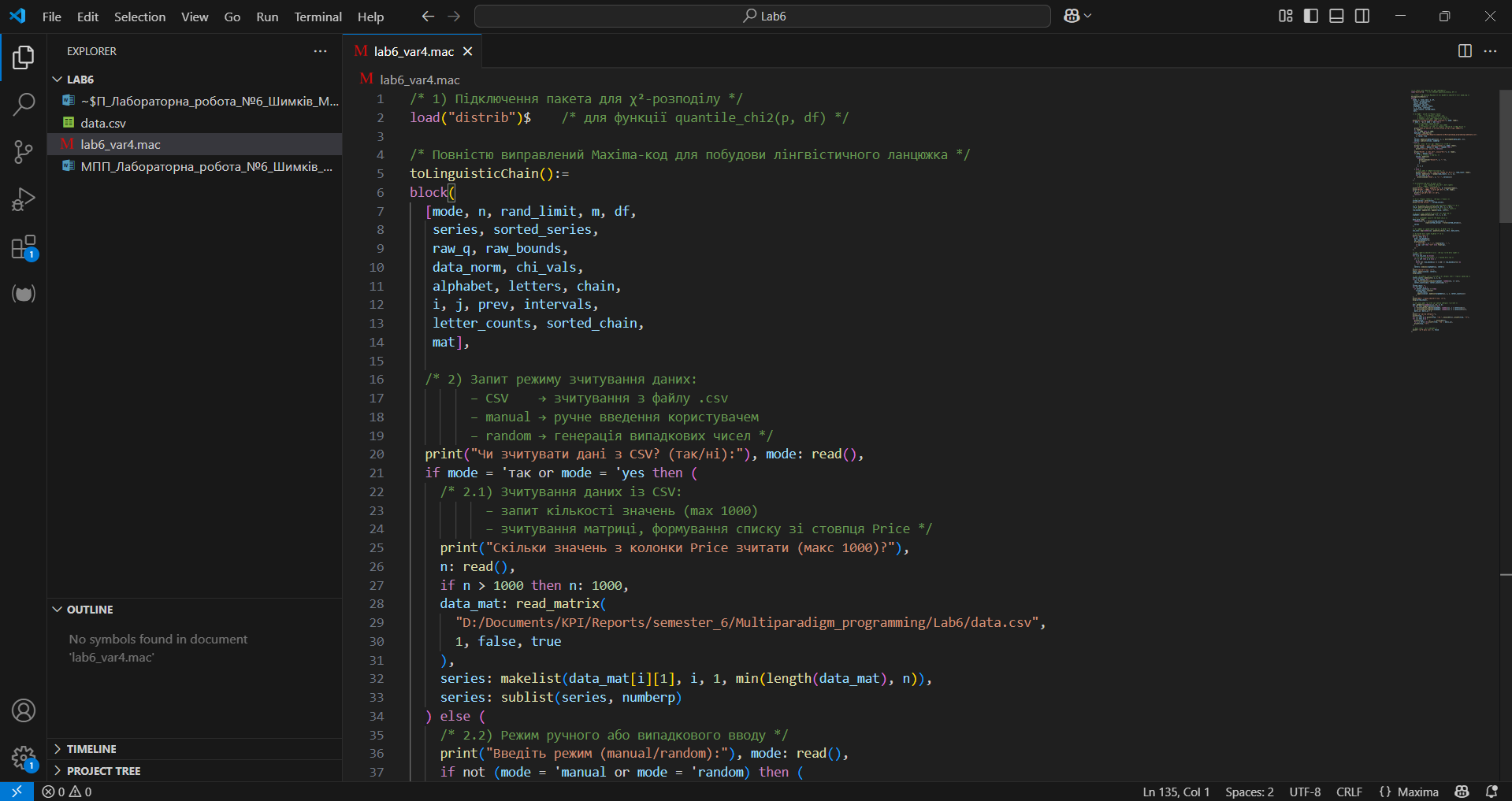
**Виконання роботи**

**Нижче наведено логічні блоки програми:**

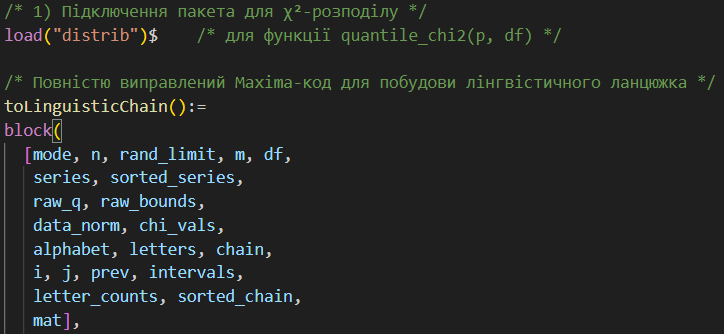
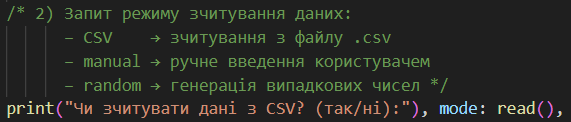
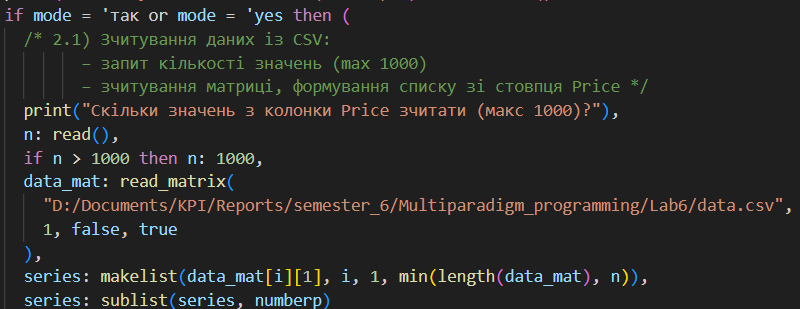
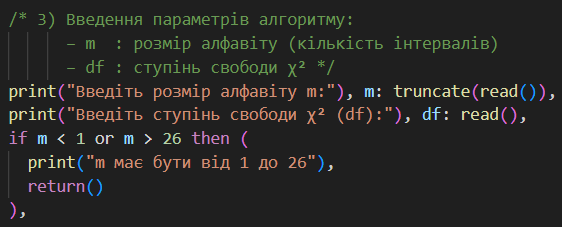
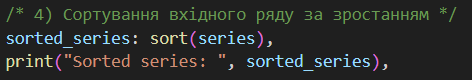
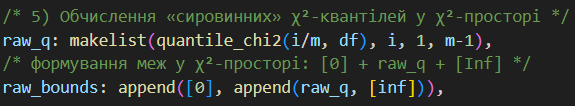
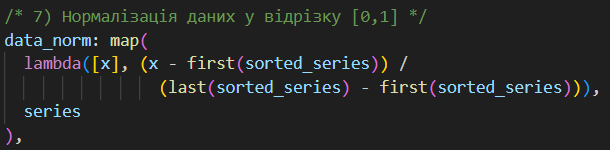
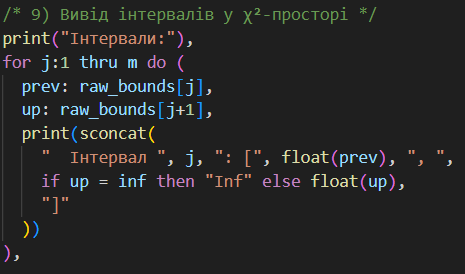
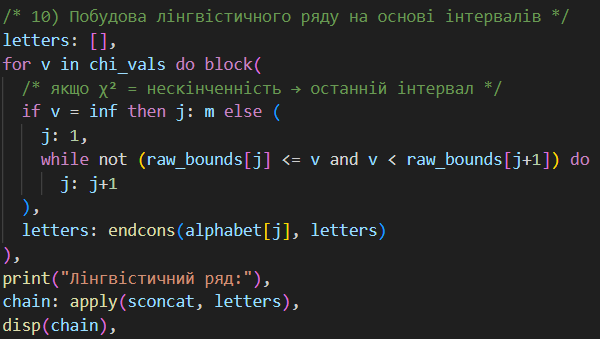
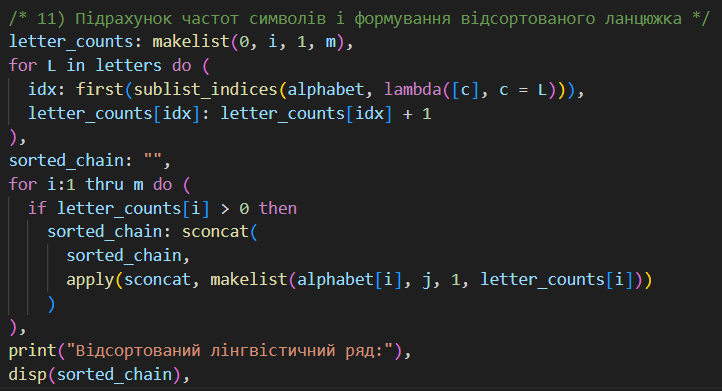
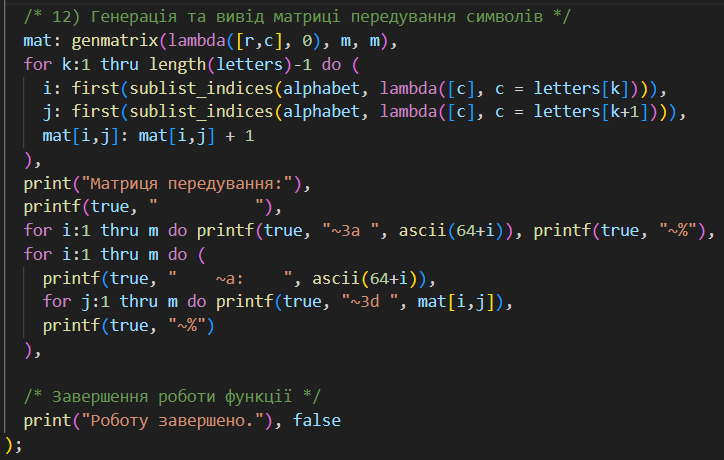
1. Підключення пакета для χ²-розподілу
2. Запит режиму зчитування даних
   1. Зчитування даних із CSV
   2. Режим ручного або випадкового вводу
3. Введення параметрів алгоритму
4. Сортування вхідного ряду за зростанням
5. Обчислення «сировинних» χ²-квантілей у χ²-просторі
6. Створення алфавіту символів для ланцюжка
7. Нормалізація даних у відрізку [0,1]
8. Мапінг нормалізованих значень у χ²-простір
9. Вивід інтервалів у χ²-просторі
10. Побудова лінгвістичного ряду на основі інтервалів
11. Підрахунок частот символів і формування відсортованого ланцюжка
12. Генерація та вивід матриці передування символів

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

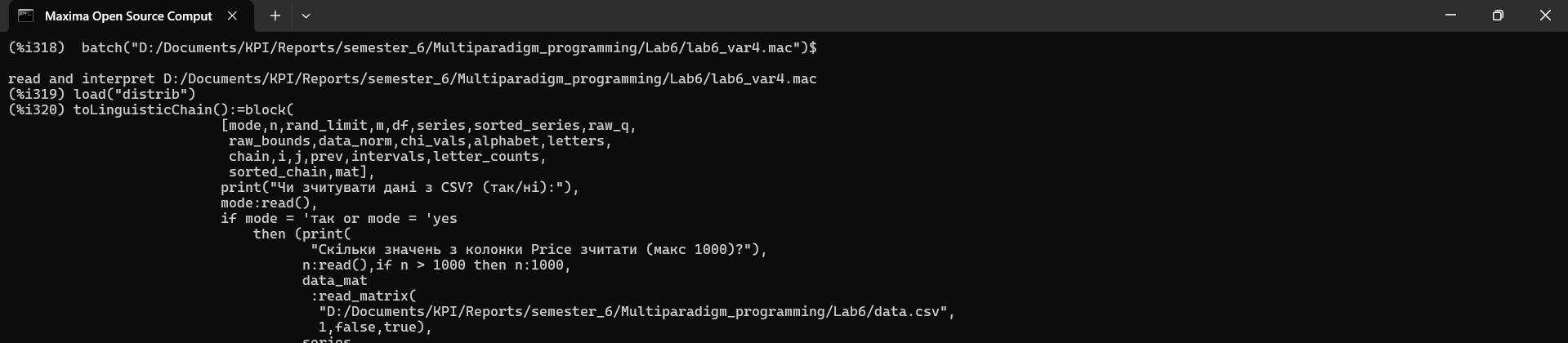
Інтерфейс Visual Studio Code for MAXIMA для запуску програми



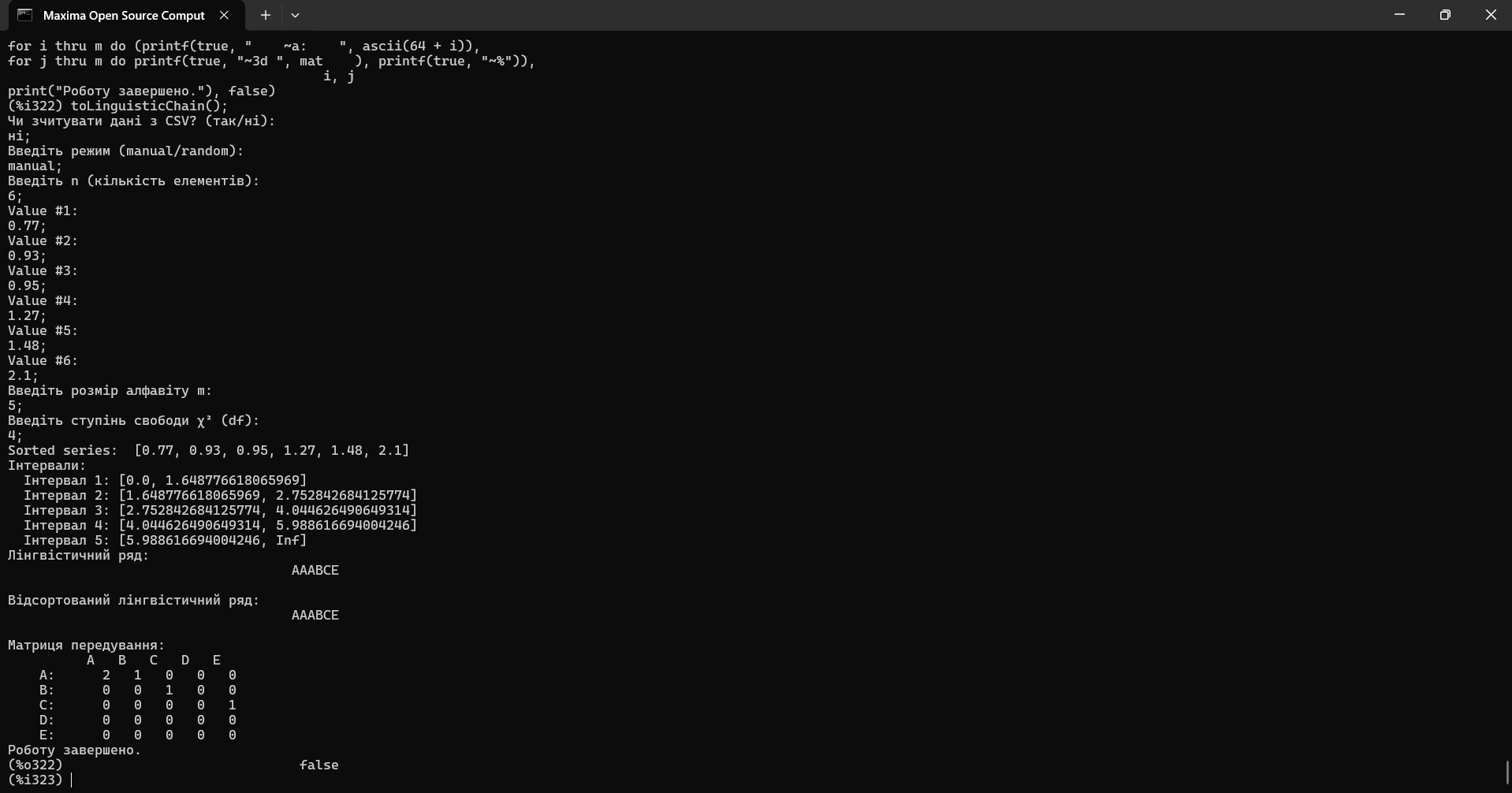
Етапи:

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

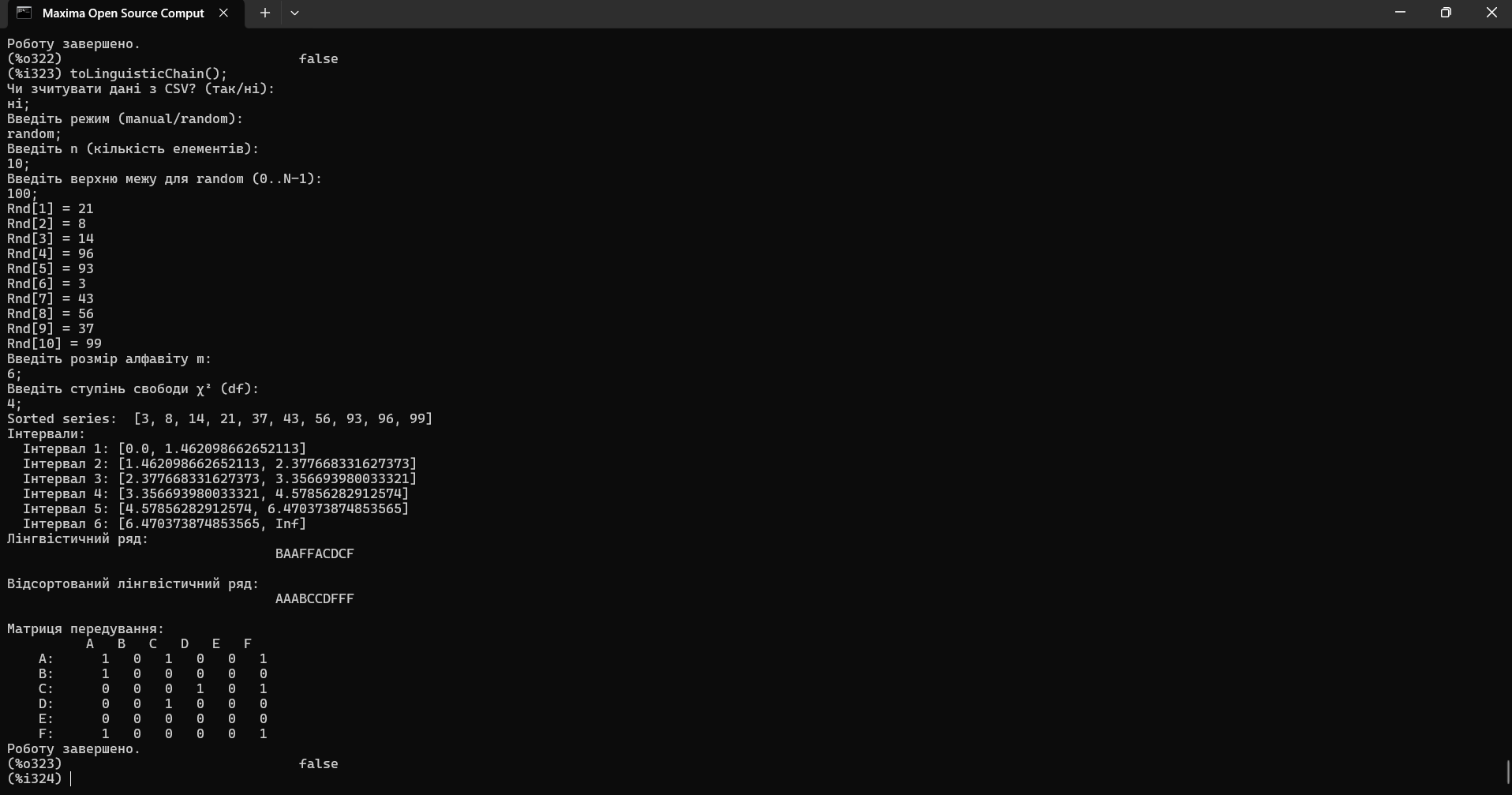
**Вхідні данні і результат виконання програми:**



Введення значень від руки:

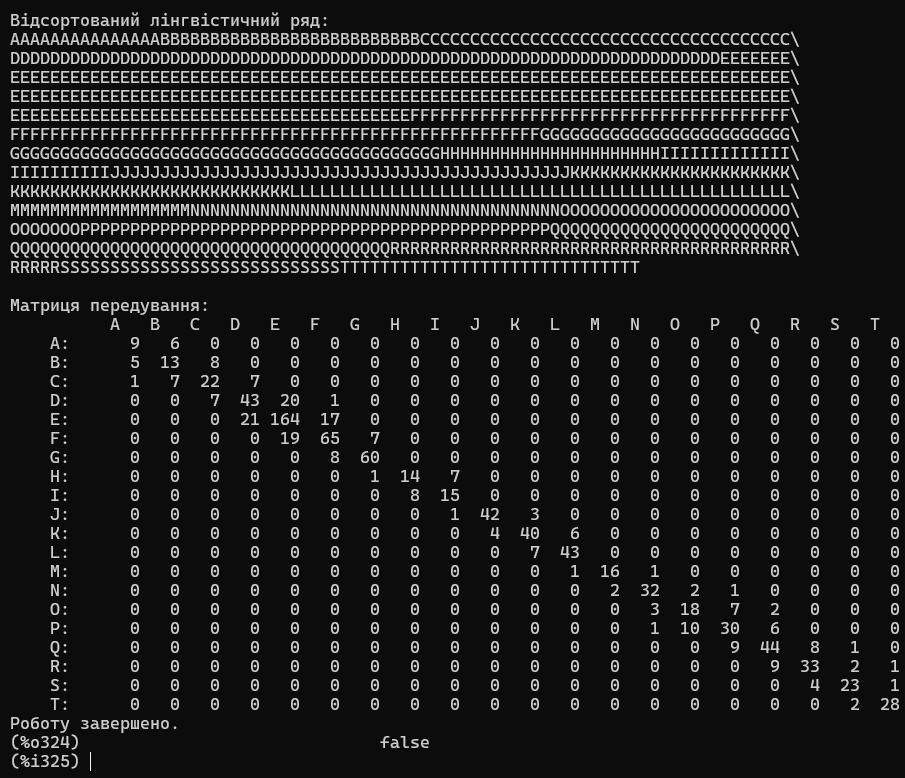
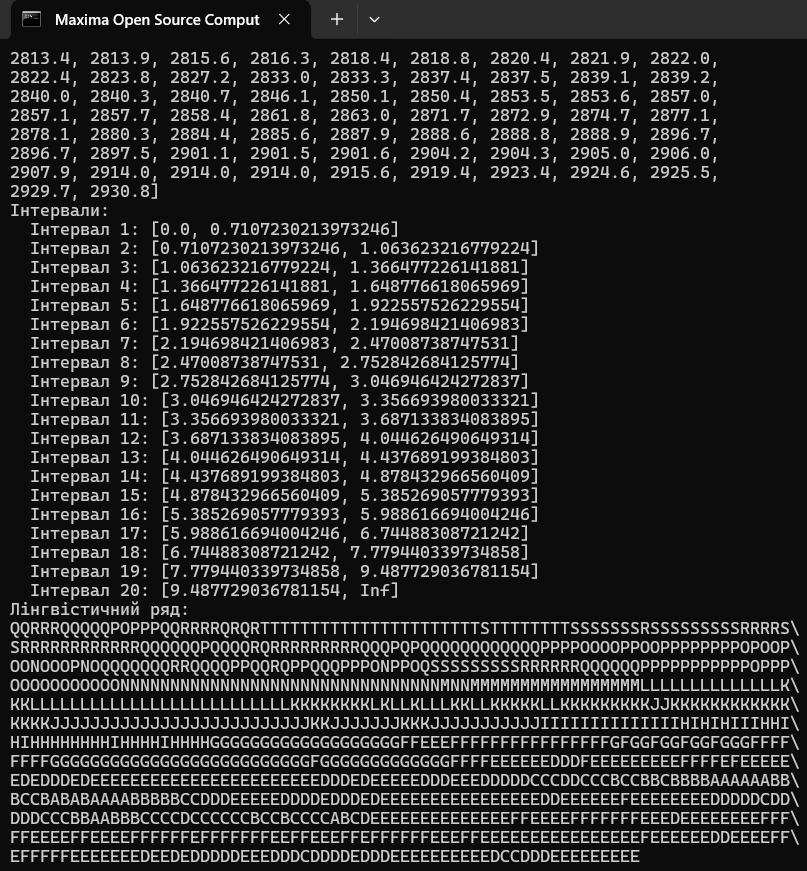


Генерація випадкових значень:



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

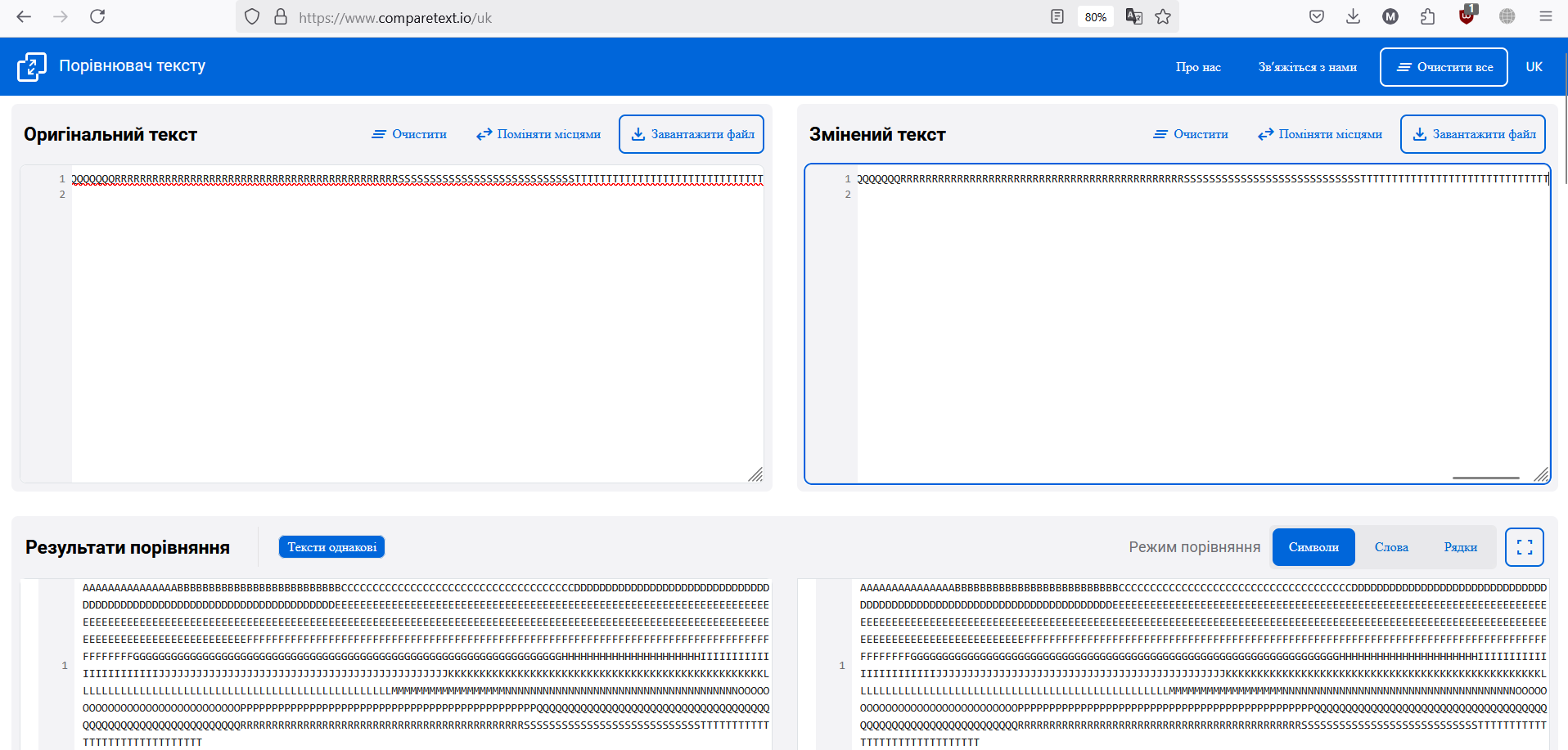




Поточний лінгвістичний ряд:

AAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLMMMMMMMMMMMMMMMMMMNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRSSSSSSSSSSSSSSSSSSSSSSSSSSSSTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

Порівняння лінгвістичних рядів лабораторної роботи 5 лабораторної роботи 6:

****

**Лінгвістичні ряди ідентичні.**

**Висновок**

У рамках лабораторної роботи №6 реалізовано програму на базі Maxima, яка:  
• Зчитує чисельний ряд, обраний розподіл ймовірностей та потужність алфавіту.  
• Сортує ряд у порядку зростання і визначає мінімальне й максимальне значення (область припустимих значень).  
• Розбиває область на інтервали відповідно до кумулятивної функції обраного розподілу, враховуючи потужність алфавіту.  
• Мапує кожне числове значення на символ алфавіту за індексом інтервалу.  
• Формує лінгвістичний рядок із отриманих символів.  
• Виводить у консоль відсортований ряд, межі інтервалів і згенерований лінгвістичний рядок.

Результати тестів (як за ручного введення, так і при випадковій генерації) доводять коректність алгоритму:  
• Відсортовані значення точно відповідають очікуваному порядку.  
• Межі інтервалів відтворюють розбиття області згідно з обраним розподілом, забезпечуючи правильні «ритми» попадання в інтервали.  
• Лінгвістичний рядок коректно відображає відповідність кожного числа своєму символу алфавіту.

Отже, завдання ЛР №6 виконане повністю й відповідає вимогам лабораторної роботи з алгебраїчного програмування.

**Додатки**

Код програми:

/\* 1) Підключення пакета для χ²-розподілу \*/

load("distrib")$    /\* для функції quantile\_chi2(p, df) \*/

/\* Повністю виправлений Maxima-код для побудови лінгвістичного ланцюжка \*/

toLinguisticChain():=

block(

  [mode, n, rand\_limit, m, df,

   series, sorted\_series,

   raw\_q, raw\_bounds,

   data\_norm, chi\_vals,

   alphabet, letters, chain,

   i, j, prev, intervals,

   letter\_counts, sorted\_chain,

   mat],

  /\* 2) Запит режиму зчитування даних:

        – CSV    → зчитування з файлу .csv

        – manual → ручне введення користувачем

        – random → генерація випадкових чисел \*/

  print("Чи зчитувати дані з CSV? (так/ні):"), mode: read(),

  if mode = 'так or mode = 'yes then (

    /\* 2.1) Зчитування даних із CSV:

          – запит кількості значень (max 1000)

          – зчитування матриці, формування списку зі стовпця Price \*/

    print("Скільки значень з колонки Price зчитати (макс 1000)?"),

    n: read(),

    if n > 1000 then n: 1000,

    data\_mat: read\_matrix(

      "D:/Documents/KPI/Reports/semester\_6/Multiparadigm\_programming/Lab6/data.csv",

      1, false, true

    ),

    series: makelist(data\_mat[i][1], i, 1, min(length(data\_mat), n)),

    series: sublist(series, numberp)

  ) else (

    /\* 2.2) Режим ручного або випадкового вводу \*/

    print("Введіть режим (manual/random):"), mode: read(),

    if not (mode = 'manual or mode = 'random) then (

      print("Невірний режим!"), return()

    ),

    print("Введіть n (кількість елементів):"), n: read(),

    if mode = 'manual then (

      /\* ручне введення n значень \*/

      series: makelist(

        block([x],

          print(sconcat("Value #", i, ": ")),

          x: read(),

          x

        ),

        i, 1, n

      )

    ) else (

      /\* генерація n випадкових значень \*/

      print("Введіть верхню межу для random (0..N-1):"), rand\_limit: read(),

      series: makelist(1 + random(rand\_limit), i, 1, n),

      for i:1 thru n do

        print(sconcat("Rnd[", i, "] = ", series[i]))

    )

  ),

  /\* 3) Введення параметрів алгоритму:

        – m  : розмір алфавіту (кількість інтервалів)

        – df : ступінь свободи χ² \*/

  print("Введіть розмір алфавіту m:"), m: truncate(read()),

  print("Введіть ступінь свободи χ² (df):"), df: read(),

  if m < 1 or m > 26 then (

    print("m має бути від 1 до 26"),

    return()

  ),

  /\* 4) Сортування вхідного ряду за зростанням \*/

  sorted\_series: sort(series),

  print("Sorted series: ", sorted\_series),

  /\* 5) Обчислення «сировинних» χ²-квантілей у χ²-просторі \*/

  raw\_q: makelist(quantile\_chi2(i/m, df), i, 1, m-1),

  /\* формування меж у χ²-просторі: [0] + raw\_q + [Inf] \*/

  raw\_bounds: append([0], append(raw\_q, [inf])),

  /\* 6) Створення алфавіту символів для ланцюжка \*/

  alphabet: makelist(ascii(64 + i), i, 1, m),

  /\* 7) Нормалізація даних у відрізку [0,1] \*/

  data\_norm: map(

    lambda([x], (x - first(sorted\_series)) /

                  (last(sorted\_series) - first(sorted\_series))),

    series

  ),

  /\* 8) Мапінг нормалізованих значень у χ²-простір \*/

  chi\_vals: map(lambda([u], quantile\_chi2(u, df)), data\_norm),

  /\* 9) Вивід інтервалів у χ²-просторі \*/

  print("Інтервали:"),

  for j:1 thru m do (

    prev: raw\_bounds[j],

    up: raw\_bounds[j+1],

    print(sconcat(

      "  Інтервал ", j, ": [", float(prev), ", ",

      if up = inf then "Inf" else float(up),

      "]"

    ))

  ),

  /\* 10) Побудова лінгвістичного ряду на основі інтервалів \*/

  letters: [],

  for v in chi\_vals do block(

    /\* якщо χ² = нескінченність → останній інтервал \*/

    if v = inf then j: m else (

      j: 1,

      while not (raw\_bounds[j] <= v and v < raw\_bounds[j+1]) do

        j: j+1

    ),

    letters: endcons(alphabet[j], letters)

  ),

  print("Лінгвістичний ряд:"),

  chain: apply(sconcat, letters),

  disp(chain),

  /\* 11) Підрахунок частот символів і формування відсортованого ланцюжка \*/

  letter\_counts: makelist(0, i, 1, m),

  for L in letters do (

    idx: first(sublist\_indices(alphabet, lambda([c], c = L))),

    letter\_counts[idx]: letter\_counts[idx] + 1

  ),

  sorted\_chain: "",

  for i:1 thru m do (

    if letter\_counts[i] > 0 then

      sorted\_chain: sconcat(

        sorted\_chain,

        apply(sconcat, makelist(alphabet[i], j, 1, letter\_counts[i]))

      )

  ),

  print("Відсортований лінгвістичний ряд:"),

  disp(sorted\_chain),

  /\* 12) Генерація та вивід матриці передування символів \*/

  mat: genmatrix(lambda([r,c], 0), m, m),

  for k:1 thru length(letters)-1 do (

    i: first(sublist\_indices(alphabet, lambda([c], c = letters[k]))),

    j: first(sublist\_indices(alphabet, lambda([c], c = letters[k+1]))),

    mat[i,j]: mat[i,j] + 1

  ),

  print("Матриця передування:"),

  printf(true, "          "),

  for i:1 thru m do printf(true, "~3a ", ascii(64+i)), printf(true, "~%"),

  for i:1 thru m do (

    printf(true, "    ~a:    ", ascii(64+i)),

    for j:1 thru m do printf(true, "~3d ", mat[i,j]),

    printf(true, "~%")

  ),

  /\* Завершення роботи функції \*/

  print("Роботу завершено."), false

);