Міністерство освіти і науки України КПІ ім. Ігоря Сікорського Факультет Інформатики та Обчислювальної Технiки

**ЗВІТ**

**Лабораторна робота №4 з дисципліни**

**«Мультипарадигменне програмування»**

|  |  |
| --- | --- |
| Перевірив:  Викладач кафедри ІПІ, ОТ  ФІОТ  Очеретяний О.К | Виконав:  Климентьєв Нікіта  гр. ІК-21 |

2025

**ЛАБОРАТОРНА 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 та матриця передування.

**Мова програмування:** Prolog.

**Хід розв’язання задачі:**

1. Чисельний ряд сортується від найменшого значення до найбільшого. Тим самим отримуємо діапазон (область) припустимих значень.

2. ОПЗ розбивbаємо на інтервали (кількість залежить від потужності обраного алфавіту) у відповідності до розподілу ймовірностей, при цьому враховуємо, що ймовірність потрапляння до інтервалу P[a,b]=F(b)-F(a).

3. Кожному числовому значенню ставиться у відповідність знак з алфавіту, який має той самий індекс, що і інтервал.

4. Після заміни отриманий ряд літер видається на вихід.

5. На основі лінгвістичного ряду будується матриця

**Обраний варіант:**

|  |  |
| --- | --- |
| **№ варіанту** | **Розподіл ймовірностей** |
| **20** | Розбиття на рівні інтервали |

**Розв`язання задачі:**

Логіка роботи розбивається на наступні етапи:

1. **Введення даних**:
   * Користувач вводить потужність алфавіту N (ціле число, N > 0), яке визначає кількість інтервалів та літер у алфавіті.
   * Користувач вводить числовий ряд — послідовність дійсних чисел, розділених пробілами, у одному рядку.
2. **Генерація алфавіту**:
   * На основі введеного N формується алфавіт із послідовних літер, починаючи з 'A'. Наприклад, для N = 3 алфавіт складається з літер {A, B, C}. Цей процес виконується предикатом alphabet\_power/2.
3. **Сортування чисел**:
   * Вхідний числовий ряд сортується для визначення мінімального (Min) та максимального (Max) значень. Сортування реалізовано предикатом sorted\_numbers/2.
4. **Розбиття на інтервали**:
   * Діапазон від Min до Max ділиться на N рівних інтервалів. Крок інтервалу обчислюється як Step = (Max - Min) / N. Наприклад, для Min = 73.05, Max = 79.44 і N = 3 інтервали будуть приблизно [73.05, 75.18], [75.18, 77.31], [77.31, 79.44]. Цей етап реалізовано предикатом intervals/4.
5. **Формування лінгвістичного ряду**:
   * Кожне число з вхідного ряду зіставляється з відповідним інтервалом. Якщо число потрапляє в i-й інтервал, йому присвоюється i-та літера алфавіту (початок з 0, тобто A для першого інтервалу).
   * Для коректної обробки верхньої межі інтервалів додано допуск плюс/мінус 0.0001 для компенсації похибок із плаваючою комою. Цей процес реалізовано предикатами assign\_letter/4 і linguistic\_sequence/4 із використанням maplist/3 для мапування чисел на літери.
6. **Побудова матриці переходів**:
   * Для лінгвістичного ряду обчислюється матриця розміром N x N, де елемент [i, j] показує кількість переходів від літери i до літери j у послідовності. Наприклад, у ряді CBCCCBCCCBABAAA перехід C -> B відбувається 3 рази.
   * Переходи підраховуються за допомогою предиката transitions/3, а матриця формується предикатом transition\_matrix/3.
7. **Виведення результатів**:
   * Виводиться лінгвістичний ряд як конкатенація літер.
   * Виводиться матриця переходів у зрозумілому форматі з заголовками (літерами алфавіту). Це реалізовано предикатами print\_matrix/2 і print\_row/3.

Програма включає відлагоджувальні виводи (наприклад, значення чисел і відповідних літер), які допомагають перевірити правильність роботи на кожному етапі.

**Структура програми**

Програма складається з набору предикатів, кожен із яких виконує специфічну функцію. Нижче наведено детальний опис структури:

* **main/0**:
  + Основний предикат, який керує виконанням програми. Організує введення даних, викликає інші предикати та обробляє помилки. Перевіряє, чи N є додатним цілим числом, і забезпечує виведення результатів.
* **alphabet\_power/2**:
  + Генерує список літер алфавіту на основі N. Використовує findall/3 для створення послідовності літер від 'A' до необхідного символу.
* **read\_numbers/1**:
  + Зчитує числовий ряд із вхідного потоку, розбиває його на числа за пробілами та повертає список чисел. Використовує split\_string/4 і number\_string/2.
* **sorted\_numbers/2**:
  + Сортує вхідний список чисел і повертає відсортований список для визначення Min і Max. Реалізовано через вбудований предикат sort/2.
* **intervals/4**:
  + Розраховує N рівних інтервалів на основі Min і Max. Використовує арифметичні операції та findall/3 для створення списку пар [A, B], де A — початок, B — кінець інтервалу.
* **assign\_letter/4**:
  + Присвоює літеру числу, визначаючи, якому інтервалу воно належить. Використовує nth0/3 для доступу до інтервалів і включає логіку для обробки верхньої межі з допуском плюс/мінус 0.0001.
* **linguistic\_sequence/4**:
  + Формує лінгвістичний ряд, застосовуючи maplist/3 до списку чисел для їх перетворення в літери за допомогою assign\_letter\_with\_debug/4.
* **assign\_letter\_with\_debug/4**:
  + Допоміжний предикат, який викликає assign\_letter/4 і виводить відлагоджувальну інформацію (число та відповідну літеру).
* **transition\_matrix/3**:
  + Будує матрицю переходів, використовуючи findall/3 для генерації рядків матриці на основі всіх можливих пар літер із алфавіту.
* **transitions/3**:
  + Підраховує кількість переходів між двома літерами у послідовності за допомогою nextto/3 і findall/3.
* **print\_matrix/2**:
  + Виводить матрицю переходів із заголовками (літерами алфавіту) і рядками, використовуючи maplist/3 для ітерації по рядках матриці.
* **print\_row/3**:
  + Виводить один рядок матриці, включаючи літеру та відповідні значення переходів, із форматуванням пробілами.
* **write\_with\_space/1**:
  + Допоміжний предикат для форматування чисел у рядку матриці з пробілами.

Програма використовує бібліотеки library(strings) і library(lists) для роботи з рядками та списками відповідно. Структура програми модульна, що дозволяє легко модифікувати або розширювати її функціонал.

**Результат розрахунку:**

Результат розрахунку з потужностю алфавіта 10 і першими 100 числами з числового ряду:

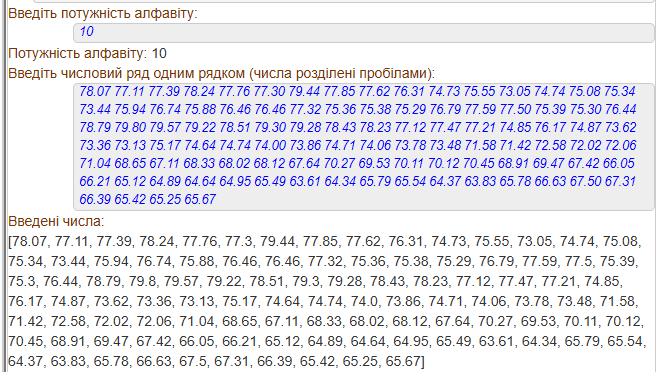


Рис. 1 – Вхідні дані

Лінгвістичний ряд: IIIJIIJIIHGHFGHHGHIHHHIHHHIIIHHHJJJJJJJJJIIIGHGGGFHGGGGGGGGEEFFFEDCCCCCEDEEEDDCBBAAAABAABBAABBCCBBBB

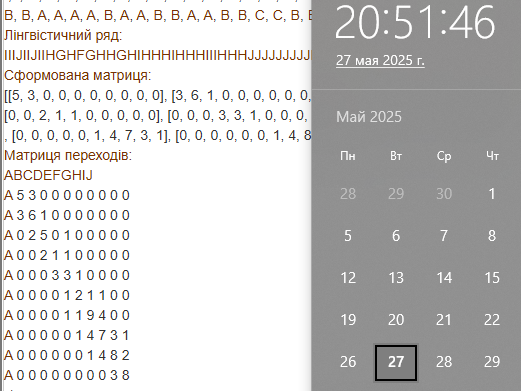


Рис. 2 – Результат розрахунку

Результат розрахунку з потужностю алфавіта 26 і всіма(5000) числами з числового ряду:

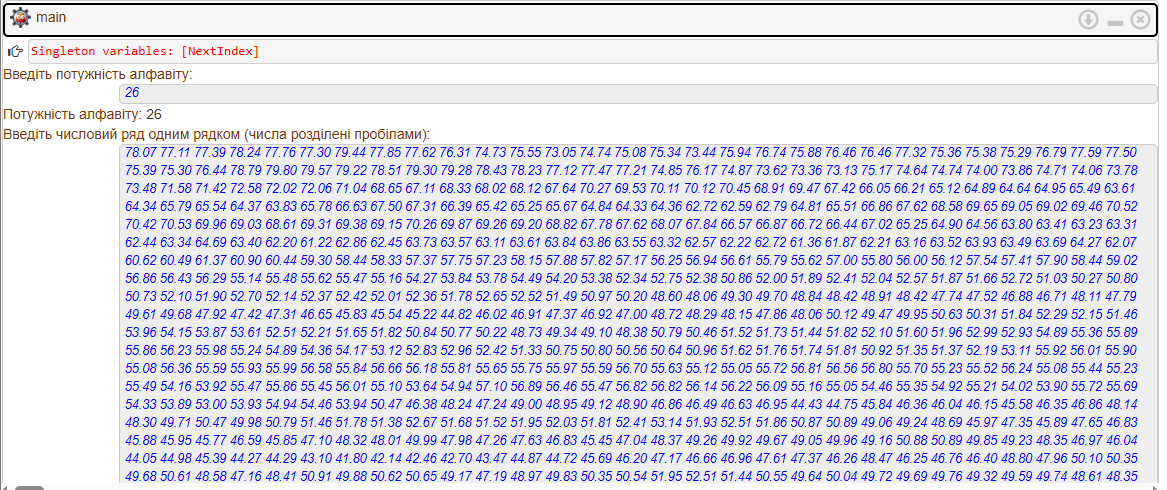


Рис. 3 – Вхідні дані

Лінгвістичний ряд: NMMNMMNMMMMMMMMMMMMMMMMMMMMMMMMMNNNNNNNNNMMMMMMMMMMMMMMMMMMLLLLLLLKLLLLLLLLLLLKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKLLLLLLLLLLLLLLLLLLLLLLLLKKKKKKKKKKKKKKKKJJKKKKKKKKKKKJKJJJKKKKKKJJJJJJJJJJJJJJJJIIIIIIIIIJJJJJIIIIIIIIIIIIIIIIIHIHIIIHHIHHHHIHIIIIIIHIIHHHHHHHHHHHHHHHHHHHHHHGGGGGGHHHHHHHHHHHHHHHIIHIIIIIIHHHHHHHHHHHHHHHIHIIIIIIIIIIIIIIIIIHHHHHHHHHHHHHIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIHGHHHHHHHGGHGGGGGGGGHHHHHHHHHHIHHIIHIIIIHHHHHHGHGHHGGGGGHHHHHHHHGHHHHHHHHHHHHHHGGGGGGGGGGGGGGGGHGHHHGHGHGHHHHHHHHHHHHHHHHHHHIIHHHHHHHHHHHHHHHHHHHHGGGGGGGHHHGGGGGGGGGGGGGFFFFFFFFFFFGGFGFFFFFFFFFFFFEEEEEEEEEEEEDDDEEEEEEEEEEDEDDDDDEDDEEEEFFFFFFFFEFFFFFFFFFFFGGGGGGGGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHHHIIHIHHHHHHHHHHHHHHGGHHHHHHHHHIHHGGGGGHHHHHHHHHHHHHIIIIIIIIIIIJIJJJJIIIJJJKJKKKKKKKKKKKKKKKKJKKKKKJKKKKKKKKKKKKKKLKKKKKKKKKKJKKKJJJJIIJJIIIIIJIIIIIIIIIIJIJJJJJJKJJJJJJJKJJIIIJJIIJIIHHHHHHHHHHHHGHHHHHIIJJJJJJJJJJJJJKKKKLLLLLLLMNNNNNNNNMNNNONNNOOOOOOOOOOOOOOOPPPPPPPPQQQQQQQQQQQRQQQQRRRRRRRRRRRRRRRRRRRSRSSSSSSSSSSSSSSSSSSSSSSSSSTTTTTTTTTTTUUTTTTTTTTSSSSSSTTTTTTTTSTTTSSSSSSSSSSSSTTSSTTTSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSTSSTTTTTTTSSSSSSTSSSSSSSSSSSSSSSSSSSSSSSSSSTTTTTTTTTTSTSSTSSTTTTTTTTTTTTSSSSSSSSSRSSSSSTSTSSSTTTSTTTTTSTTTSSSSSSSSSSSTSTTTTTTUUUUTTUUTTTTTTTTTTTSSSSSSSTSSSSSSSSSSSSSSSSSSSSSRRRRRRRRRSSSSSRRRSRRRRRRRSRRRRRSSRRRRRSSSSRRRRRRRRRRRRQRRRSSSSRSSTTTTSSSSSSTTSSTTTTTTTTTTTTTTUUUUUUUUUUUUUUUUTTTTTTTTTTTTTTTTTTTTTTTTSSTTSSSSTSSSSSTTTTTTTTTTTTSTTSSSSSTSSSSSSTSSSSTTTUUTUTTTTTSTTTTTTTTTSTTUUUUUTTTTUTTTTTTUUTTTUUTTTTTTTSSSSSSSSRRSSSRRRRRQRQRRRQQPPPPPPPQQQQQQQRRRQQQRRSSSSSSSSSTTTTTTTTTUUUUUVUUUUUUUUVVVUVVVVWVVVWWVVVVWWVVWWWWVVVVWVVVWVVVVUVUUUUUUUUTTTTTTTTTTTTTTTTTTTTTTTTSSSSSSSSRRSSTSSSTTTTSTTSSSSSSSSTTTTTTUTTTSTTTTSTTTTSTTTTTTSSSRRRRRRSRRSTTSTUTTTTTUTTTTUTTTTTSSSSTTTSSSRRSSTUUUUUUUUUUUUUUUUUUUTTTTTTSSSSTTTTTUVUUUUUTUUTUTUUUTTTTTTTTTTUUSTVVVWVVVVVVVVVVVVVWVVVVUUUUUUUUUUTUTSTTUUTUUUUUTTTTSSRRRRRRRRRRRRRRRRQQQQQQQQQQQQQQPQQPQQPQQQQQPPPPPPPPPPPPPPPPOOOOONOOONOOOPPOOOOOOONONOONNONONOOOOOOOONONNNNNNMNNNNNNNNMNMMMMMMMMMMLMMMMMMMMMNNNNNNNNMMMMMMMMMMMMMMMMMLLLLMMMNMMMNNNNMMMMMLLLMMLMMMLLLLLMMMMNNNNNNNOPOOOOOOOOOOOOOOOOOOOOONNNNNNNNNNNNMNNNNNNNNNMMMNMNNMMMLMMLLLLLMMMLLLMMLMMMMMMNNNNNNNNMMMMMMMMMMMLLLLLMMMNMNNMMNMMMMNNNMMMMMMMNNMMMMMMNNNMMMMMLLLLKLLLLLKKKKLLLLLLKKLLLLKKKLLLLLLLMMMMLLLMMLMMMMMMLLKLLLLKKKKKKJJJJJKKKKLLLLLLLKLLLLLLLLLLLLKLLKKKJJJJJJJIIJJJJIIIIIHHHHHHHHHIIHIIIHHIIIHHHIIHIIHHHHGGGFGGGGGGGGGGGFGGFFGGGGGGGGGGGGGGGHHGGGGGGGGGGGGHHHGFFFFFGGGGGGGHGGGFGGGHIIIHIHHHHIIIIIJJJJKJKKKJJJKKLLLKLMMMNOOOPPQQQRSRRSRQQPPQQQRRRSSSSTTUTTTVTTTTTTTTTUUUVVVWVWVWWWXWXYYZZZYYZZZZYYYYXYYXXYXXXYXXXYWVVWWWXWXXXXWVVVVVVWVVVUTTTTUUTUUTTTTTTSSSSSSRRRRRSRRRRRSRSSSSRRRRQRRRQRQQQQQQQQPPPPOOPPPPPPPPPOOOPPPPPPPQQQQQQQQQQPPPPPPPPQPOPPOPPOPPPQQQQQPPPPPPPPPPPPPPOPPOONNOONONNNNMMNNMMMNNMMNNNNMMMMMMMMMMMMLLLLLLLLLLLLLLLLLLLLMMMMMMMMMMMMMMMMMMMMMMMMLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLKKKKKKKKKKLLLLKLKKKKKLLLKKLLLLLLKKKKKJJJJJJJJJJKJJJJJJJJJJJJJJJIIJJJJJJIJIIIIIIIIHIIIIHIIIIIJJJJJJKKKKJKJJJJJKKKKKKKJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJKKJJJJJJJKKJKKKKKKLLLLLLLLLLMMLLMMMMMMMNMMMMMMMMMMMMMMMMMMMMMMMLMMMLLLLLLLLLKKKLLLLLLLLLLLLLLLLLLLLLLLLLLLMMMLLLLMMMMLLLLLLKLKKKKKKKKKKJJJKKKKJJJJJJKKKJJKJJJJJJJJJJJJJKKKKKKKKKKKKKKKKJKJJJKJJJJJJIIIIIIIJJJJJJIJJIIIIIIIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJJJJJJJJJKKKKKKKKKKKJKKJJKKKKKKLKKKKKKKKKKKKKKKJKJJJJJJJJJJJIIJIJJJJJJJJJJJIIJJJJIJJJIIIIIIIIIIIIHHHHHHHHHHHHHHHHHHHHHIIIIIIIIHHHHIIIIIIIIIIIIIIIIIIIIIIIIIHIHHHHHHHHGGGGGGGGGGGGGGGGGGHGGGGGGGGGGGGGFFFFFFFFGGGGGFFFFFFFFFGGGGGGGGGGGGFGGGGGGGHGHHHHHHHHHHHHHHHHHHHHGGGGGGGGGGGGFGGFFGFFFFGGFFFFGGGGGGGGGGFGFFFFFFFFFFFFFFFFFFFFFFFFFEEEEEEEEEEFFEEEEEEEEEFFFFFFFFFFFFFFFFFEFFFEEEEEEEEEEEEEEEEEEDDEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDDEDDDDDDDDDDDDDDDDDDEDEDDEEEEEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDCCDDDDCCCCCCCCCCCCCCDCCCCCCCCCCCDDDDCCDCCDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDEEEDDDDDDDDEEDDDDDDDDDDDDDDDDDDDDDDDDCDCDCCCCCCCCCCCCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCCDDDDDDDDCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCDDDDDDDDDCDCDDDDDDDDDDCCCCCDDDDDDDDCCCCCCCCCCCCCCCCCCCCCCBBBBCBCCCBCBBBBBBBBBBBBBBBBBBBBCCCCCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCBCBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDDDDDCCCCCCCDDDCCDDCDCCCCCCCCCCCCCCCCDDDDDCCDCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCDCCCCCCCCCCDDDDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCDDDCCCCCCCCCCCCCCCDDCCDDDDDDDDEEEEEEEEEEEEEEEEEEEDDEDEDEEEEEDEEEEEEEDDDDEEDDDDDEEEEEEEEEEEEEEEEEEEDDDDDDDEEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDEEEEDDDDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCDCCCCDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDCCDDDDDDDCCCCCCCCCCCCCCCCCCDCCCCCCCDCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAAAABBBABAAABBBBBBBBBBBBBBBBBBBBBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

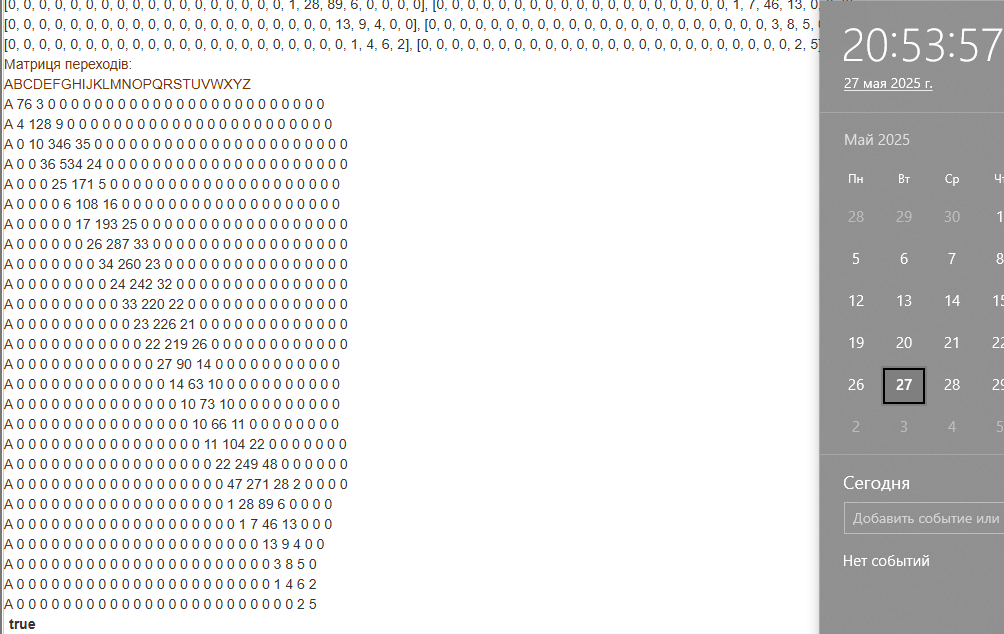


Рис. 4 – Результат розрахунку

**Лістінг програмного тексту:**

program.pl:

#lang racket

;; Функція для зчитування чисел з окремих рядків

(define (read-numbers)

(let loop ([numbers '()])

(define line (read-line))

(cond [(or (eof-object? line) (string=? line ""))

(reverse numbers)]

[else

(define num (string->number line))

(if num

(loop (cons num numbers))

(error "Некоректне число у введенні"))])))

;; Функція для визначення індексу інтервалу для числа

(define (get-interval-index x min-val width alphabet-size)

(if (zero? width)

0

(min (floor (/ (- x min-val) width)) (sub1 alphabet-size))))

;; Функція для відображення чисел на літери за рівними інтервалами

(define (map-to-letters numbers alphabet-size)

(define min-val (apply min numbers))

(define max-val (apply max numbers))

(define width (if (= alphabet-size 1)

0

(/ (- max-val min-val) alphabet-size)))

(for/list ([x numbers])

(define index (inexact->exact (get-interval-index x min-val width alphabet-size)))

(integer->char (+ 65 index)))) ; 65 — ASCII-код для 'A'

;; Функція для побудови матриці передування

(define (build-matrix chain alphabet-size)

;; Ініціалізація матриці з окремими векторами для кожного рядка

(define matrix (for/vector ([i (in-range alphabet-size)])

(make-vector alphabet-size 0)))

;; Побудова матриці передування

(for ([i (in-range (sub1 (string-length chain)))])

(define prev (char->integer (string-ref chain i)))

(define next (char->integer (string-ref chain (add1 i))))

(define row (- prev 65)) ; 'A'=65 -> 0, 'B'=66 -> 1, 'C'=67 -> 2

(define col (- next 65))

(vector-set! (vector-ref matrix row) col

(add1 (vector-ref (vector-ref matrix row) col))))

matrix)

;; Функція для виведення матриці з позначками літер

(define (print-matrix matrix alphabet-size)

(printf " ")

(for ([i (in-range alphabet-size)])

(printf "~a " (integer->char (+ 65 i))))

(newline)

(for ([i (in-range alphabet-size)])

(printf "~a " (integer->char (+ 65 i)))

(for ([j (in-range alphabet-size)])

(printf "~a " (vector-ref (vector-ref matrix i) j)))

(newline)))

;; Головна функція для координації програми

(define (main)

(define alphabet-size (string->number (read-line)))

(when (or (< alphabet-size 1) (> alphabet-size 26))

(error "Потужність алфавіту має бути від 1 до 26"))

(define numbers (read-numbers))

(when (null? numbers)

(error "Числа не введено"))

(define chain (list->string (map-to-letters numbers alphabet-size)))

(define matrix (build-matrix chain alphabet-size))

(printf "Лінгвістичний ланцюжок:\n~a\n" chain)

(printf "Матриця передування:\n")

(print-matrix matrix alphabet-size))

;; Запуск програми

(main)