

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Лабораторна робота № 6
з дисципліни “Мультипарадигмене програмування”

Виконав:
Студент групи ІО-23
Швед А. Д.

Київ - 2025

Завдання

За допомогою системи алгебраїчного програмування реалізувати перетворення чисельного ряду до лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів.

Вхідні дані

Чисельний ряд, вид розподілу ймовірностей, потужність алфавіту.

Вихідні дані

Лінгвістичний ряд.

Мова програмування

MAXIMA.

Хід роботи

Програма реалізує перетворення числового ряду у лінгвістичний ланцюжок з наступним побудуванням матриці передування.

Правила

На початку програма визначає вхідні дані: числовий список `numbers` та набір лінгвістичних символів `alphabet`. Далі визначаються мінімальне (`min_val`) та максимальне (`max_val`) значення. Різницю між цими значеннями ділять на кількість символів алфавіту `n`, після чого формується список інтервалів — кожен інтервал відповідає певному символу алфавіту.

Після цього кожен елемент початкового числового ряду перетворюється на лінгвістичний символ за допомогою функції `number_to_symbol`. Для кожного значення обчислюється індекс відповідного інтервалу (заокругленням донизу), і воно зіставляється з відповідною літерою.

Через значне сповільнення `wxMaxima` через рендеринг довгого лінгвістичного ряду у символьному режимі результат був перетворений на простий текст та виведений у окремий файл.

Реалізація виконана в середовищі `Maxima` — системі символьних обчислень із семантикою, подібною до `Lisp`. Завдяки використанню високорівневих конструкцій, таких як `block` (для оголошення локальних змінних і послідовного виконання коду), `makelist` та `map` (для створення і трансформації списків), а також вбудованих функцій `sort`, `lmin`, `lmax`, `floor` і роботи зі списками, програма залишається лаконічною й зрозумілою. Однією з особливостей підходу є поєднання символьного й числового програмування, що дає змогу ефективно працювати з числами високої точності, динамічно формувати структури даних і створювати анонімні функції для гнучких перетворень.

Результати виконання

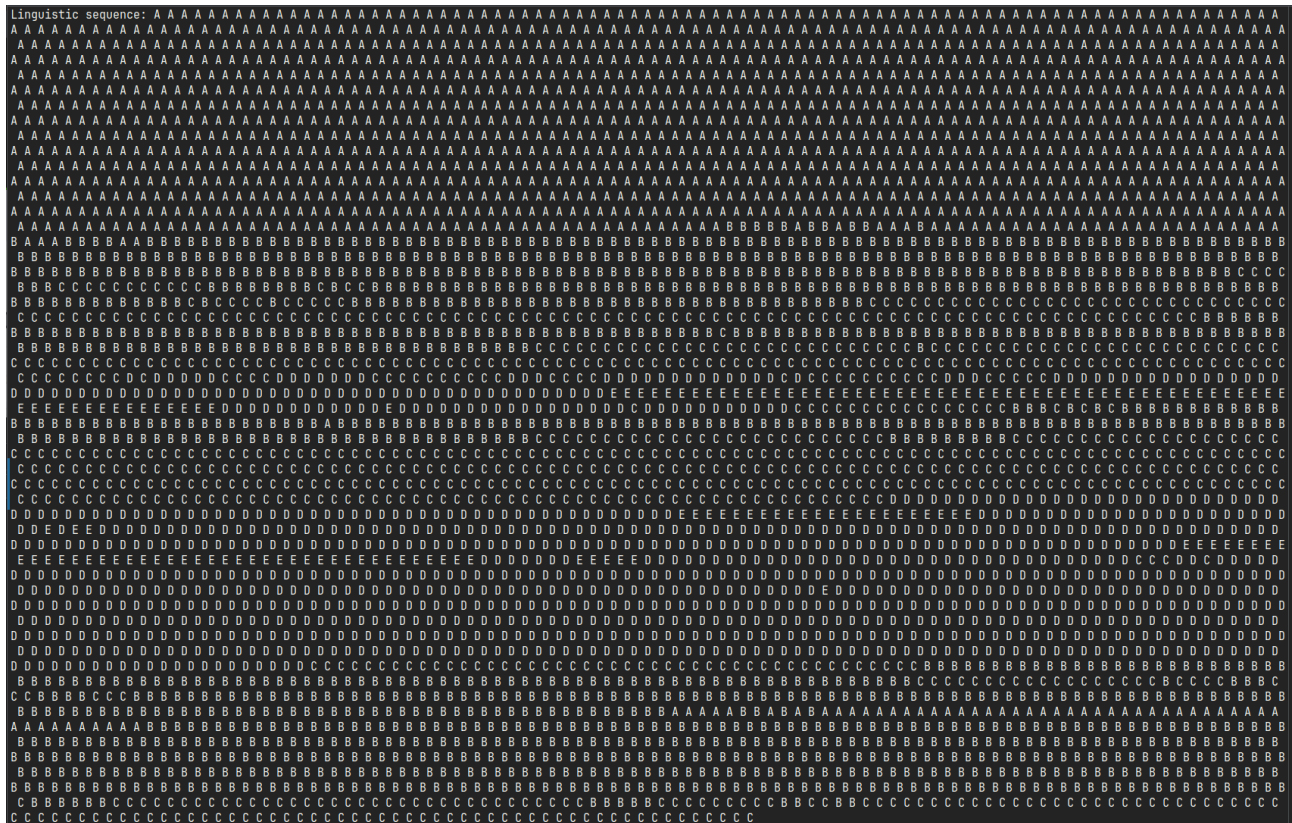


Figure 1: Перший числовий ряд В-С-D-E-F-Brent Oil Futures Historical Data (5000 значень - 5 СИМВОЛІВ)

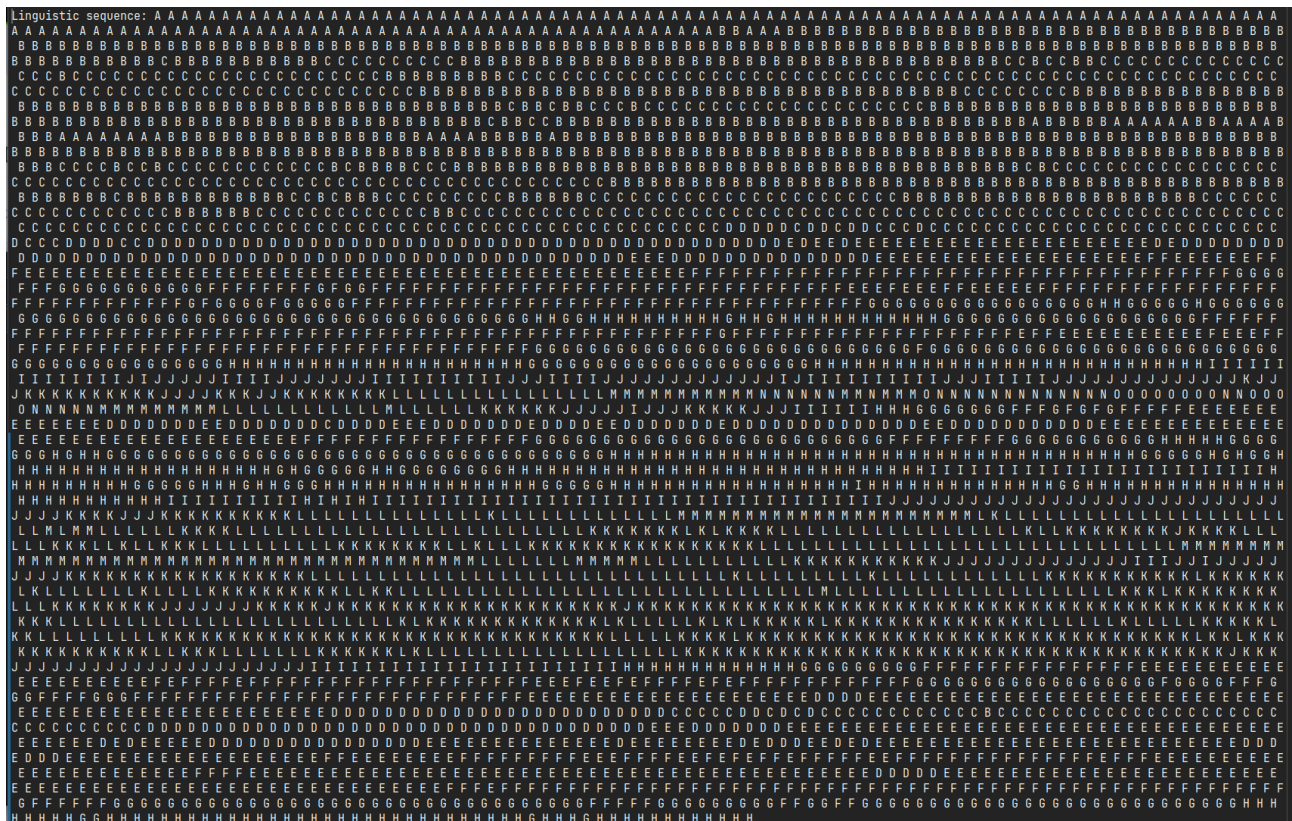


Figure 2: Перший числовий ряд В-С-D-E-F-Brent Oil Futures Historical Data (5000 значень - 15 символів)

Лістинг коду

main.mac

```
1  /* Reading from file function */
2  read_number_series(filename) := block(
3    [stream, line, num, numbers : []],
4    stream : openr(filename),
5    while (line : readline(stream)) # false do (
6      if stringp(line) and line # "" then (
7        num : parse_string(line),
8        push(num, numbers)
9      )
10   ),
11   close(stream),
12   return(numbers)
13 );
14
15
16 /* === Input initialisation === */
17 numbers : read_number_series("C:/Users/Andrii/Documents/data.txt")$
18 alphabet : ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O"]$
19
20
21 /* === Interval calculation === */
22 n : length(alphabet)$
23 min_val : lmin(numbers)$
24 max_val : lmax(numbers)$
25 interval_length : (max_val - min_val) / n$
26 intervals : makelist(
27   [ min_val + (i-1)*interval_length,
28     min_val + i*interval_length ],
29   i, 1, n
30 )$
31
32
33 /* === Linguistic sequence building === */
34 number_to_symbol(x) := block(
35   [idx],
36   idx : floor((x - min_val) / interval_length) + 1,
37   if idx > n then idx : n,
38   alphabet[idx]
39 )$
40 linguistic_sequence : map(number_to_symbol, numbers)$
41
42
43 /* === ВИВІД РЕЗУЛЬТАТІВ === */
44 with_stdout("C:/Users/Andrii/Documents/res_2.txt",
45   block(
46     printf(true, "Linguistic sequence: "),
47     printf(true, "~a", sconcat(apply(concat, map(lambda([x], sconcat(x, " ")),
48       linguistic_sequence))))
49   )
50 )$
```