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 №1**

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІС-23

Шимків Мирослав

Київ 2025

**ЛАБОРАТОРНАЯ РОБОТА №1**

**Завдання:** на процедурній мові програмування реалізувати перетворення чисельного ряду до лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів з подальшою побудовою матриці передування.

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

**Вихідні дані**: лінгвістичний ряд та матриця передування.

**Мова програмування**: **Фортран** або COBOL

**Варіант 24**

(завдання 4 відповідно)

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

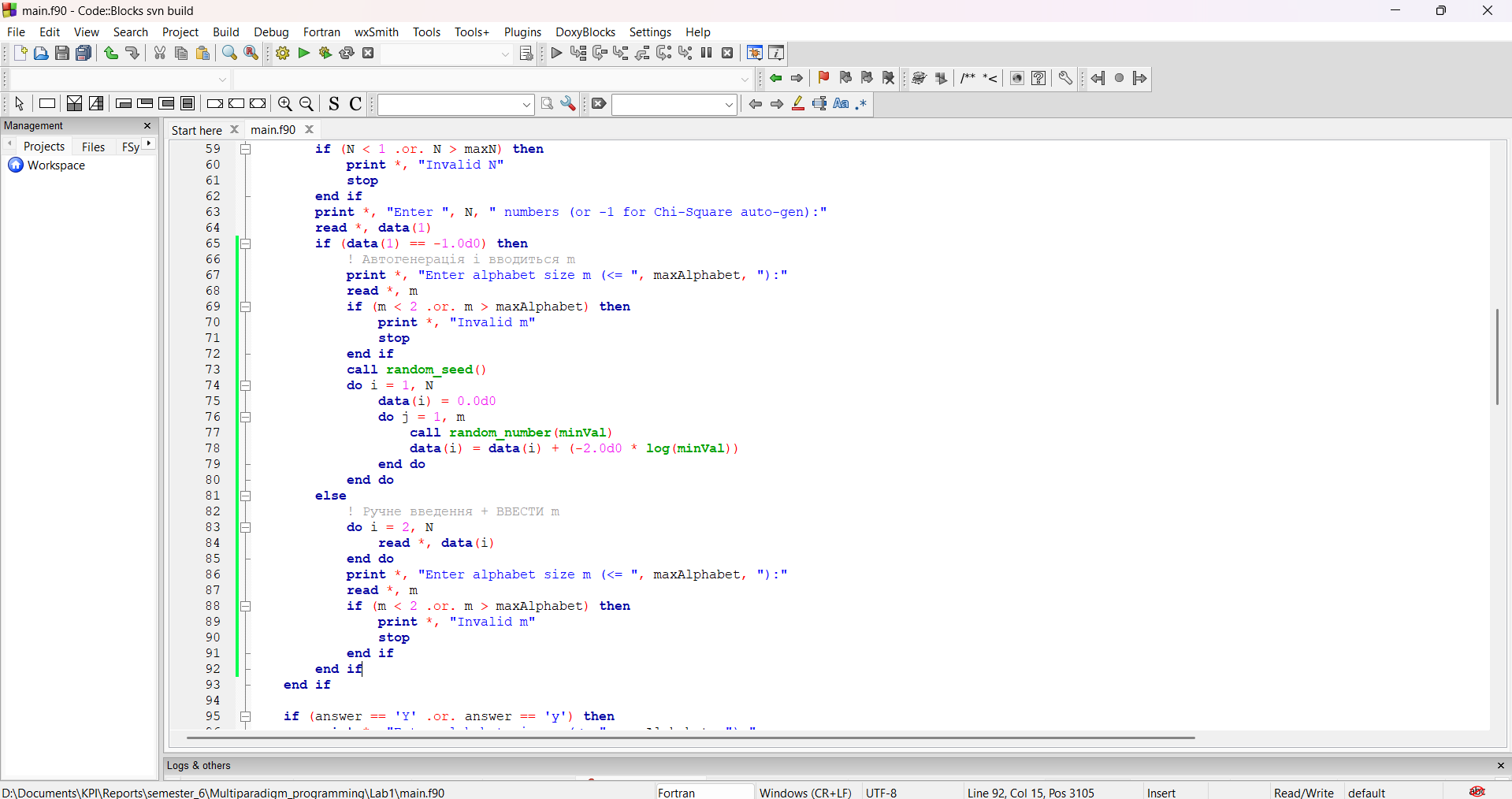
**Виконання роботи**

**Нижче наведено логічні блоки програми:**

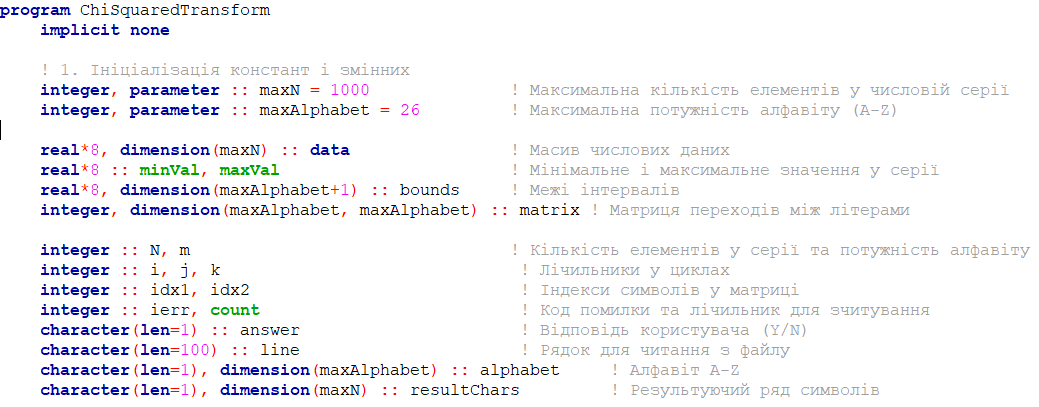
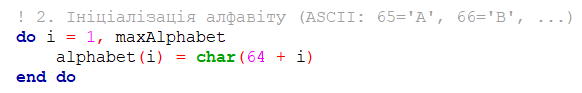
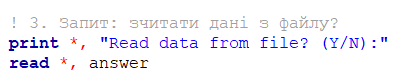
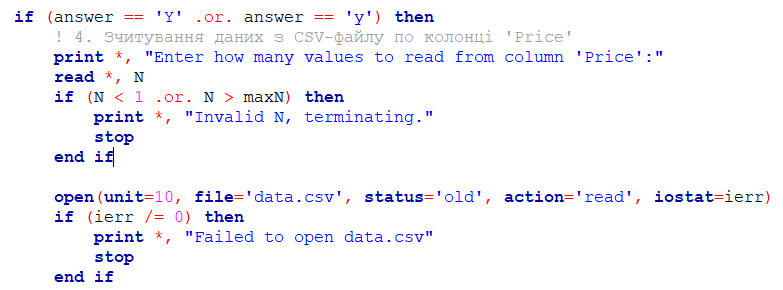
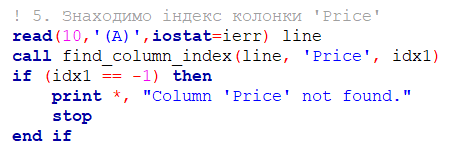
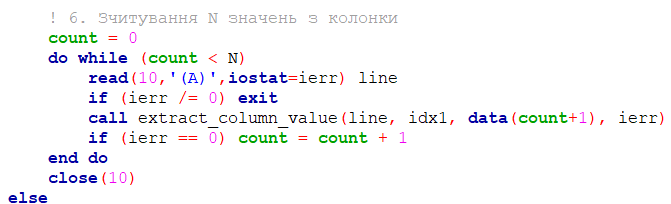
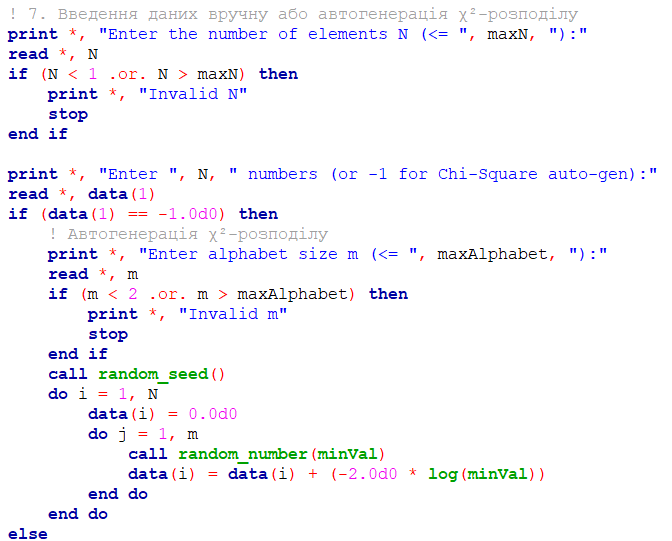
1. Ініціалізація констант і змінних.
2. Ініціалізація алфавіту (ASCII: 65='A', 66='B', ...).
3. Запит: зчитати дані з файлу.
4. Зчитування даних з CSV-файлу по колонці 'Price'.
5. Знаходимо індекс колонки 'Price'.
6. Зчитування N значень з колонки.
7. Введення даних вручну або автогенерація χ²-розподілу.
8. Якщо дані зчитувались з файлу — питаємо m окремо.
9. Сортування масиву data за зростанням (бульбашкове сортування).
10. Визначення min і max значення.
11. Розрахунок меж інтервалів (рівномірне розбиття).
12. Перетворення чисел у символи згідно інтервалів.
13. Ініціалізація матриці переходів нулями.
14. Заповнення матриці переходів (частоти переходів між літерами).
15. Вивід відсортованої серії чисел.
16. Вивід лінгвістичного рядка.
17. Вивід матриці переходів.

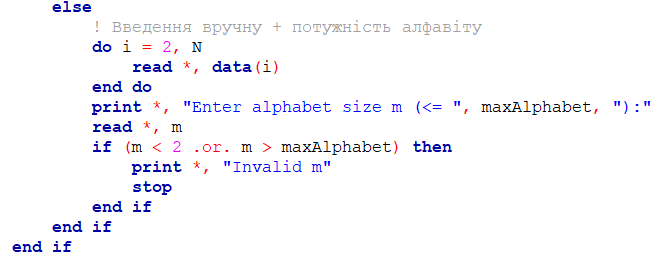
**Давайте розглянемо код для кожного пункту, в якому наведено коментарі українською мовою для кращого розуміння:**

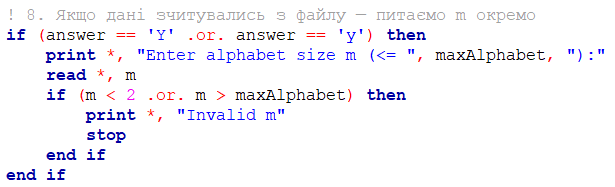
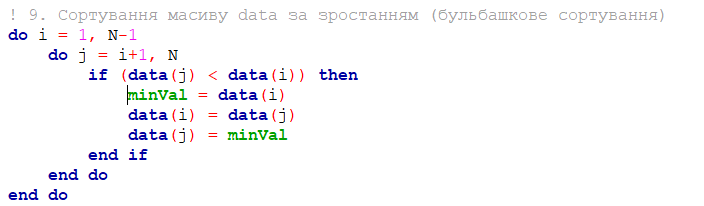
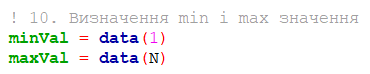
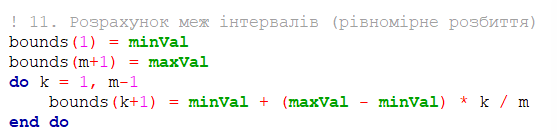
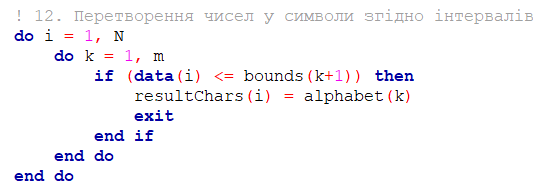
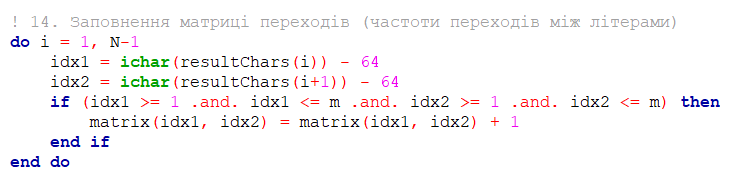
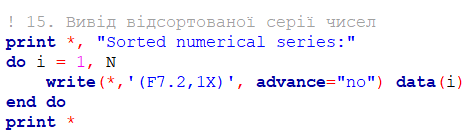
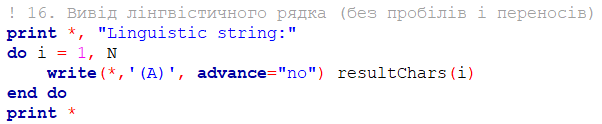
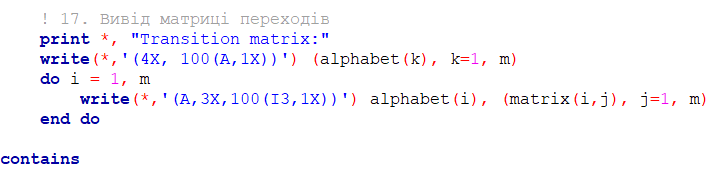
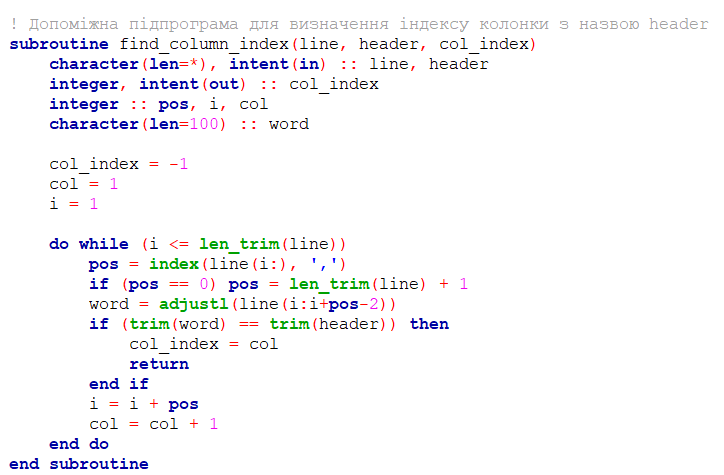
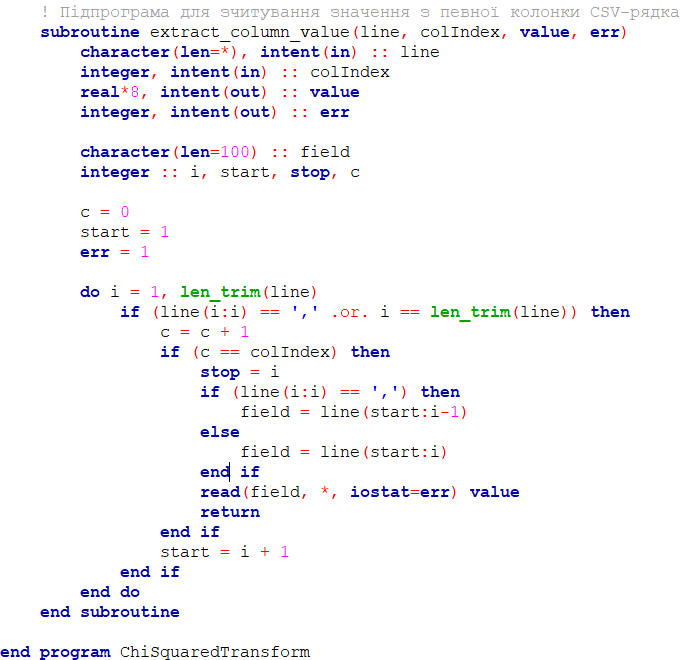
Інтерфейс Code::Blocks IDE for Fortran для запуску програми



Етапи:

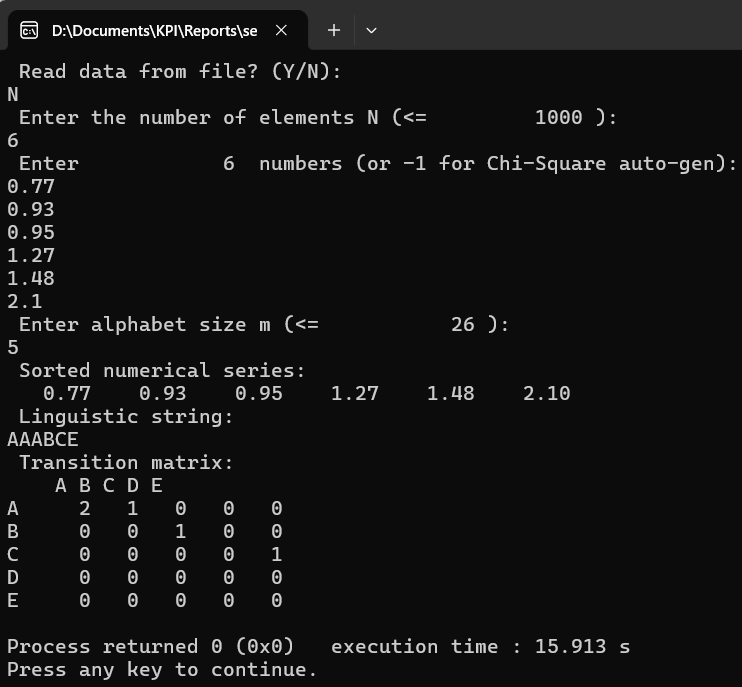
1. 
2. 
3. 
4. 
5. 
6. 
7. 



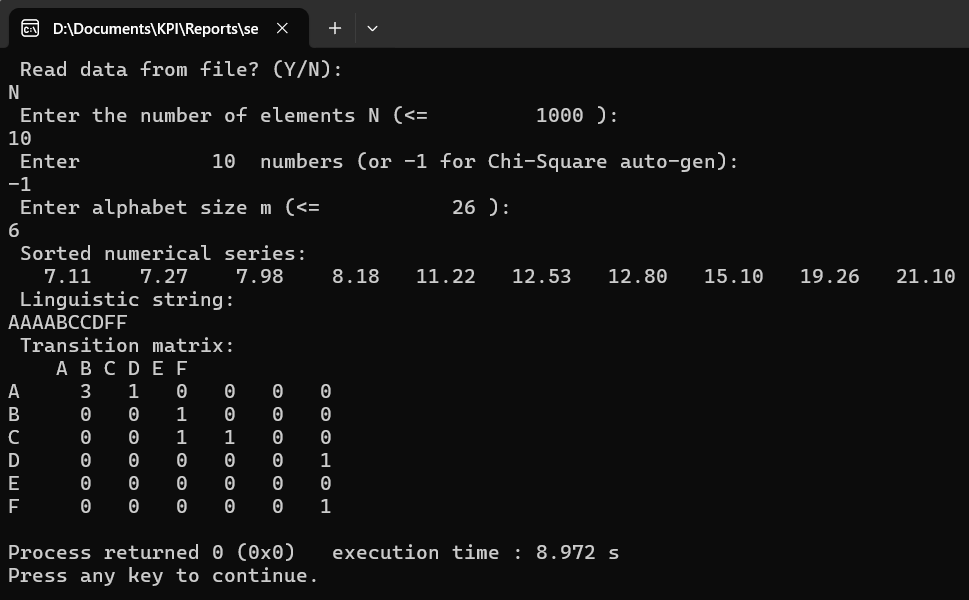
1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 

**Вхідні данні і результат виконання програми:**

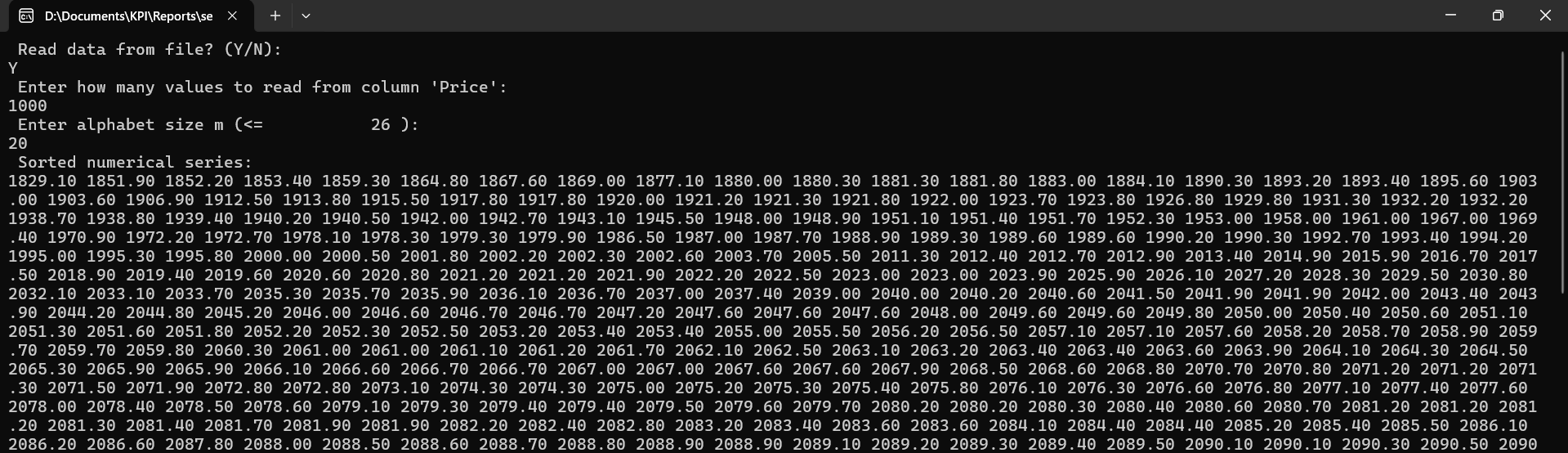
Введення значень від руки:



Генерація випадкових значень:



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





Лінгвістичний ряд:

AAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLMMMMMMMMMMMMMMMMMMNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRSSSSSSSSSSSSSSSSSSSSSSSSSSSSTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

**Висновок**

У рамках лабораторної роботи реалізовано програму на мові Фортран, яка:

* сортує вхідний числовий ряд у порядку зростання;
* розбиває діапазон на задану кількість інтервалів і перетворює кожне значення в символ алфавіту;
* формує лінгвістичний рядок та обчислює матрицю частот переходів між символами;
* коректно працює як із ручним введенням даних, так і з автоматично згенерованою χ²-послідовністю.

Результати запусків підтверджують правильність алгоритму: відсортовані значення відповідають очікуванням, символи розподілені за інтервалами рівномірно, а матриця переходів правильно відображає кількість появ кожної пари символів. Таким чином, завдання виконане повністю й відповідає вимогам ЛР.

**Додатки**

Код програми:

program ChiSquaredTransform

implicit none

! 1. Ініціалізація констант і змінних

integer, parameter :: maxN = 1000 ! Максимальна кількість елементів у числовій серії

integer, parameter :: maxAlphabet = 26 ! Максимальна потужність алфавіту (A-Z)

real\*8, dimension(maxN) :: data ! Масив числових даних

real\*8 :: minVal, maxVal ! Мінімальне і максимальне значення у серії

real\*8, dimension(maxAlphabet+1) :: bounds ! Межі інтервалів

integer, dimension(maxAlphabet, maxAlphabet) :: matrix ! Матриця переходів між літерами

integer :: N, m ! Кількість елементів у серії та потужність алфавіту

integer :: i, j, k ! Лічильники у циклах

integer :: idx1, idx2 ! Індекси символів у матриці

integer :: ierr, count ! Код помилки та лічильник для зчитування

character(len=1) :: answer ! Відповідь користувача (Y/N)

character(len=100) :: line ! Рядок для читання з файлу

character(len=1), dimension(maxAlphabet) :: alphabet ! Алфавіт A-Z

character(len=1), dimension(maxN) :: resultChars ! Результуючий ряд символів

! 2. Ініціалізація алфавіту (ASCII: 65='A', 66='B', ...)

do i = 1, maxAlphabet

alphabet(i) = char(64 + i)

end do

! 3. Запит: зчитати дані з файлу?

print \*, "Read data from file? (Y/N):"

read \*, answer

if (answer == 'Y' .or. answer == 'y') then

! 4. Зчитування даних з CSV-файлу по колонці 'Price'

print \*, "Enter how many values to read from column 'Price':"

read \*, N

if (N < 1 .or. N > maxN) then

print \*, "Invalid N, terminating."

stop

end if

open(unit=10, file='data.csv', status='old', action='read', iostat=ierr)

if (ierr /= 0) then

print \*, "Failed to open data.csv"

stop

end if

! 5. Знаходимо індекс колонки 'Price'

read(10,'(A)',iostat=ierr) line

call find\_column\_index(line, 'Price', idx1)

if (idx1 == -1) then

print \*, "Column 'Price' not found."

stop

end if

! 6. Зчитування N значень з колонки

count = 0

do while (count < N)

read(10,'(A)',iostat=ierr) line

if (ierr /= 0) exit

call extract\_column\_value(line, idx1, data(count+1), ierr)

if (ierr == 0) count = count + 1

end do

close(10)

else

! 7. Введення даних вручну або автогенерація χ²-розподілу

print \*, "Enter the number of elements N (<= ", maxN, "):"

read \*, N

if (N < 1 .or. N > maxN) then

print \*, "Invalid N"

stop

end if

print \*, "Enter ", N, " numbers (or -1 for Chi-Square auto-gen):"

read \*, data(1)

if (data(1) == -1.0d0) then

! Автогенерація χ²-розподілу

print \*, "Enter alphabet size m (<= ", maxAlphabet, "):"

read \*, m

if (m < 2 .or. m > maxAlphabet) then

print \*, "Invalid m"

stop

end if

call random\_seed()

do i = 1, N

data(i) = 0.0d0

do j = 1, m

call random\_number(minVal)

data(i) = data(i) + (-2.0d0 \* log(minVal))

end do

end do

else

! Введення вручну + потужність алфавіту

do i = 2, N

read \*, data(i)

end do

print \*, "Enter alphabet size m (<= ", maxAlphabet, "):"

read \*, m

if (m < 2 .or. m > maxAlphabet) then

print \*, "Invalid m"

stop

end if

end if

end if

! 8. Якщо дані зчитувались з файлу — питаємо m окремо

if (answer == 'Y' .or. answer == 'y') then

print \*, "Enter alphabet size m (<= ", maxAlphabet, "):"

read \*, m

if (m < 2 .or. m > maxAlphabet) then

print \*, "Invalid m"

stop

end if

end if

! 9. Сортування масиву data за зростанням (бульбашкове сортування)

do i = 1, N-1

do j = i+1, N

if (data(j) < data(i)) then

minVal = data(i)

data(i) = data(j)

data(j) = minVal

end if

end do

end do

! 10. Визначення min і max значення

minVal = data(1)

maxVal = data(N)

! 11. Розрахунок меж інтервалів (рівномірне розбиття)

bounds(1) = minVal

bounds(m+1) = maxVal

do k = 1, m-1

bounds(k+1) = minVal + (maxVal - minVal) \* k / m

end do

! 12. Перетворення чисел у символи згідно інтервалів

do i = 1, N

do k = 1, m

if (data(i) <= bounds(k+1)) then

resultChars(i) = alphabet(k)

exit

end if

end do

end do

! 13. Ініціалізація матриці переходів нулями

matrix = 0

! 14. Заповнення матриці переходів (частоти переходів між літерами)

do i = 1, N-1

idx1 = ichar(resultChars(i)) - 64

idx2 = ichar(resultChars(i+1)) - 64

if (idx1 >= 1 .and. idx1 <= m .and. idx2 >= 1 .and. idx2 <= m) then

matrix(idx1, idx2) = matrix(idx1, idx2) + 1

end if

end do

! 15. Вивід відсортованої серії чисел

print \*, "Sorted numerical series:"

do i = 1, N

write(\*,'(F7.2,1X)', advance="no") data(i)

end do

print \*

! 16. Вивід лінгвістичного рядка (без пробілів і переносів)

print \*, "Linguistic string:"

do i = 1, N

write(\*,'(A)', advance="no") resultChars(i)

end do

print \*

! 17. Вивід матриці переходів

print \*, "Transition matrix:"

write(\*,'(4X, 100(A,1X))') (alphabet(k), k=1, m)

do i = 1, m

write(\*,'(A,3X,100(I3,1X))') alphabet(i), (matrix(i,j), j=1, m)

end do

contains

! Допоміжна підпрограма для визначення індексу колонки з назвою header

subroutine find\_column\_index(line, header, col\_index)

character(len=\*), intent(in) :: line, header

integer, intent(out) :: col\_index

integer :: pos, i, col

character(len=100) :: word

col\_index = -1

col = 1

i = 1

do while (i <= len\_trim(line))

pos = index(line(i:), ',')

if (pos == 0) pos = len\_trim(line) + 1

word = adjustl(line(i:i+pos-2))

if (trim(word) == trim(header)) then

col\_index = col

return

end if

i = i + pos

col = col + 1

end do

end subroutine

! Підпрограма для зчитування значення з певної колонки CSV-рядка

subroutine extract\_column\_value(line, colIndex, value, err)

character(len=\*), intent(in) :: line

integer, intent(in) :: colIndex

real\*8, intent(out) :: value

integer, intent(out) :: err

character(len=100) :: field

integer :: i, start, stop, c

c = 0

start = 1

err = 1

do i = 1, len\_trim(line)

if (line(i:i) == ',' .or. i == len\_trim(line)) then

c = c + 1

if (c == colIndex) then

stop = i

if (line(i:i) == ',') then

field = line(start:i-1)

else

field = line(start:i)

end if

read(field, \*, iostat=err) value

return

end if

start = i + 1

end if

end do

end subroutine

end program ChiSquaredTransform