Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до виконаної лабораторної роботи №3

з дисципліни

“Чисельні методи”

на тему:

***«Розв’язування системи лінійних алгебраїчних рівнянь методом***

***Гауса. Схема Жордана-Гауса»***

Виконав

студент групи *ОІ-11сп*

*Вальчевський П. В.*

Викладач:

*Сенета М. Я.*

Львів – 2023

## Лабораторна робота № 3

**Тема роботи:** *«Розв’язання системи лінійних алгебраїчних рівнянь Методом Гауса. Схема Жордана-Гауса»*

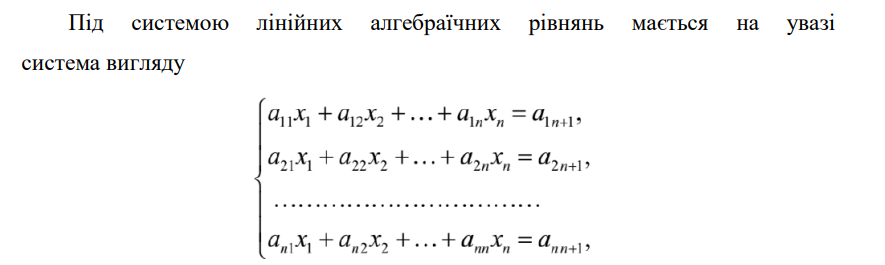
**Мета роботи:***набути вмінь та навичок використання прямих методів розв’язання систем лінійних алгебраїчних рівнянь із застосуванням систем комп’ютерної математики; вивчити і засвоїти Методи Гауса і Жордана–Гауса розв’язування СЛАР.*

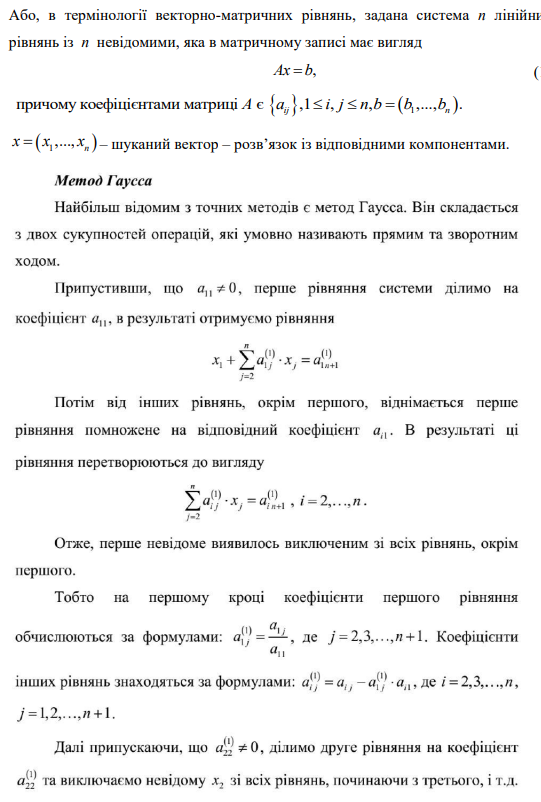
***Порядок виконання роботи***

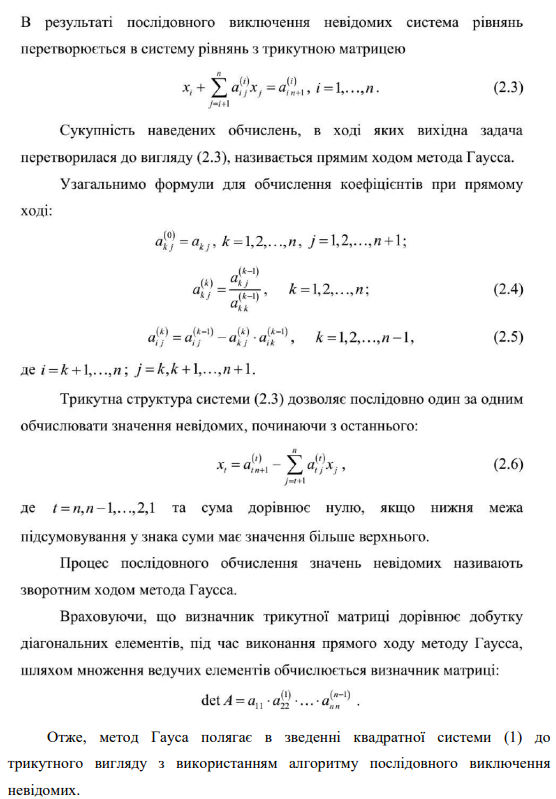
1. 1. Попереднє опрацювання теоретичного матеріалу.
2. 2. Опрацювання типового навчального завдання (прикладів).
3. 3. Створення проекту для виконання індивідуального завдання.
4. 4. Оформити звіт для захисту лабораторної роботи за зразком:

* назва роботи;
* мета роботи;
* короткі теоретичні відомості;
* алгоритм розв’язання задачі;
* лістинг програми з коментарями;
* результати виконання;
* висновки.

1. 5. Захист лабораторної роботи
2. **Короткі теоретичні відомості.**

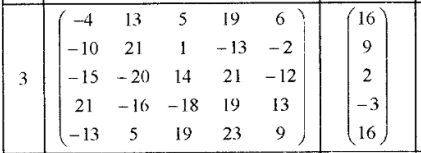
****

****

****

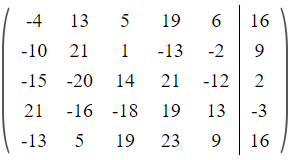
1. **Умова завдання.**

* задану СЛАР розв’язати методами Гауса та Жордана-Гауса
* знайти аналітичний розв’язок СЛАР
* навести алгоритми розв’язання системи вручну
* розробити програму для виконання завдання
* доповнити текст програми таким чином, щоб у результатах виконання програми було відображено інформацію: група, ПІБ студента, номер ЛР і варіанту завдання

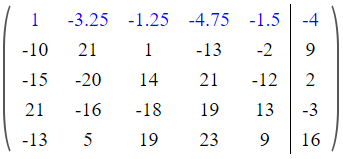
1. **Номер варіанту – 3**
2. ****

**Алгоримт розв’язання завдання.**

1. Аналіз кожного з методів згідно теоретичних методів;
2. Проектування кожного методів розв’язку СЛАР та їх допоміжних методів для реалізації;
3. Реалізація кожного методу та його тестування;
4. Проектування програми виконання з урахуванням виключень;
5. Реалізації програми виконання.
6. Повторне тестування раніше розроблених методів та програми тестування;
7. **Виконане завдання вручну (методом Гауса-Жордано).**
8. Початкова матриця та вектор у матричному вигляді:



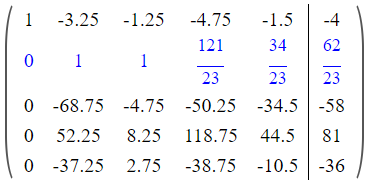
1-ий рядок ділимо на -4:



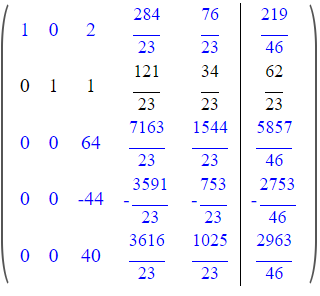
до 2 рядка додаємо 1 рядок, помножений на 10; до 3 рядка додаємо 1 рядок, помножений на 15; від 4 рядка віднімаємо 1 рядок, помножений на 21; до 5 рядка додаємо 1 рядок, помножений на 13:



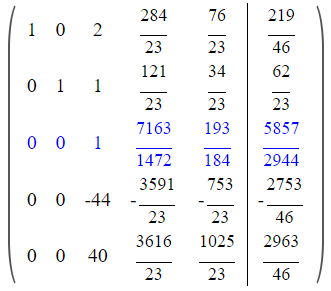
2-й рядок ділимо на -11.5:



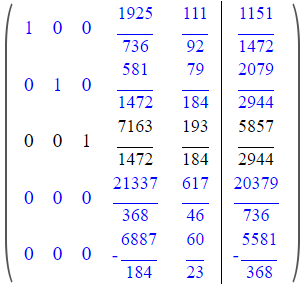
до 1 рядка додаємо 2 рядок, помножений на 3.25; до 3 рядка додаємо 2 рядок, помножений на 68.75; від 4 рядка віднімаємо 2 рядок, помножений на 52.25; до 5 рядка додаємо 2 рядок, помножений на 37.25:



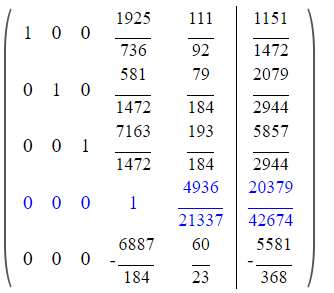
3-й рядок ділимо на 64:



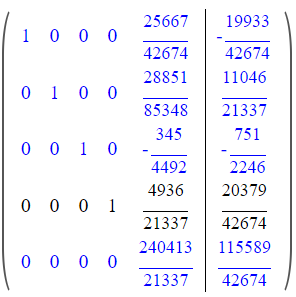
від 1 рядка віднімаємо 3 рядок, помножений на 2; від 2 рядка віднімаємо 3 рядок, помножений на 1; до 4 рядка додаємо 3 рядок, помножений на 44; від 5 рядка віднімаємо 3 рядок, помножений на 40:



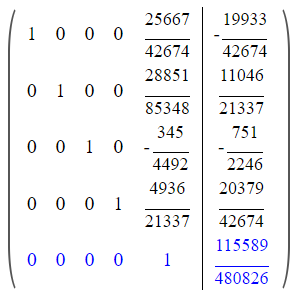
4-й рядок ділимо на :



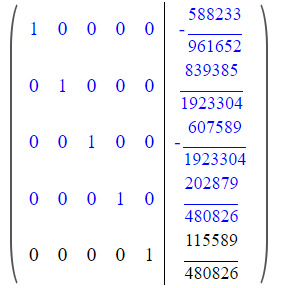
від 1 рядка віднімаємо 4 рядок, помножений на ; від 2 рядка віднімаємо 4 рядок, помножений на ; від 3 рядка віднімаємо 4 рядок, помножений на ; до 5 рядка додаємо 4 рядок, помножений на :



5-й рядок ділимо на :



від 1 рядка віднімаємо 5 рядок, помножений на ; від 2 рядка віднімаємо 5 рядок, помножений на ; до 3 рядка додаємо 5 рядок, помножений на ; від 4 рядка віднімаємо 5 рядок, помножений на :



вийшла одинична матриця – розв’язок знайдено (підставивши у початкові рівняння розв’язок усі рівняння будуть рівні нулеві при умові, що усі коефіцієнти були перенесені у ліву сторону рівняння), система розв’язку виглядає так:

**Код програмної реалізації.**

*# Метод Гауса.*def gauss(matrixA=list, vectorB=list) -> None:  
 n = len(matrixA) *# Кількість рядків у матриці* x = [0.0] \* n *# Вектор розв'язку* for k in range(n - 1):  
 print("\nЕтап", k + 1, ":")  
  
 *# Виведення початкового стану матриці A і вектора b* print("Матриця A:")  
 for i in range(n):  
 for j in range(n):  
 print("{:7.3f}".format(matrixA[i][j]), end=" ")  
 print(" |", vectorB[i])  
  
 for i in range(k + 1, n):  
 factor = matrixA[i][k] / matrixA[k][k]  
 for j in range(k, n):  
 matrixA[i][j] -= factor \* matrixA[k][j]  
 vectorB[i] -= factor \* vectorB[k]  
  
 *# Виведення оновленого стану матриці A і вектора b після елімінації Гауса* print("\nОновлена матриця A та вектор b після етапу", k + 1, ":")  
 for i in range(n):  
 for j in range(n):  
 print("{:7.3f}".format(matrixA[i][j]), end=" ")  
 print(" |", vectorB[i])  
  
 *# Зворотний хід* for i in range(n - 1, -1, -1):  
 x[i] = vectorB[i]  
 for j in range(i + 1, n):  
 x[i] -= matrixA[i][j] \* x[j]  
 x[i] /= matrixA[i][i]  
  
 print("\nЗнайдені розв'язки:")  
 for i in range(n):  
 print("x[{}] = {:.3f}".format(i, x[i]))  
  
*# Метод Гауса-Жордана.*def gauss\_jordan\_with\_steps(matrixA=list, vectorB=list) -> None:  
 n = len(matrixA)  
  
 print("Початкова матриця A:")  
 print\_matrix(matrixA)  
 print("Початковий вектор B:")  
 print\_vector(vectorB)  
 print("-------------------------------")  
  
 *# Прямий хід методу Гаусса* for i in range(n):  
 pivot = matrixA[i][i]  
 print(  
 f"Ділимо рядок {i + 1} на {pivot} для отримання одиниці в головному діагональному елементі A[{i + 1}][{i + 1}]")  
 for j in range(i, n):  
 matrixA[i][j] /= pivot  
 vectorB[i] /= pivot  
  
 print(f"Матриця A після кроку {i + 1}:")  
 print\_matrix(matrixA)  
 print(f"Вектор B після кроку {i + 1}:")  
 print\_vector(vectorB)  
 print("-------------------------------")  
  
 for k in range(n):  
 if k != i:  
 factor = matrixA[k][i]  
 print(  
 f"Віднімаємо {factor} \* рядок {i + 1} від рядку {k + 1} для зроблення нуля під головним діагональним елементом A[{k + 1}][{i + 1}]")  
 for j in range(i, n):  
 matrixA[k][j] -= factor \* matrixA[i][j]  
 vectorB[k] -= factor \* vectorB[i]  
  
 print(f"Матриця A після кроку {i + 1}:")  
 print\_matrix(matrixA)  
 print(f"Вектор B після кроку {i + 1}:")  
 print\_vector(vectorB)  
 print("-------------------------------")  
  
*# Вивід матриці для Методу Гауса-Жордана.*def print\_matrix(matrix):  
 for row in matrix:  
 for element in row:  
 print(f"{element:.3f}", end=" ")  
 print()  
  
*# Вивід вектора для Методу Гауса-Жордана.*def print\_vector(vector):  
 for element in vector:  
 print(f"{element:.3f}")  
  
*# Заміна ряду, якщо діагональний елемент рівний нулю.*def move\_rows\_if\_diagonal\_zero(matrixA=list, vectorB=list) -> None:  
 n = len(matrixA)  
  
 for i in range(n):  
 if matrixA[i][i] == 0:  
 *# Знаходимо індекс рядка, де на головній діагоналі 0* j = i + 1  
 while j < n and matrixA[j][i] == 0:  
 j += 1  
  
 *# Переміщуємо рядок j на позицію i в матриці matrix\_a* matrixA[i], matrixA[j] = matrixA[j], matrixA[i]  
  
 *# Переміщуємо відповідний елемент вектора vector\_b* vectorB[i], vectorB[j] = vectorB[j], vectorB[i]  
  
*# Функція використання методів.*def use\_method(key=int) -> None:  
 matrixA = [  
 [-4, 13, 5, 19, 6],  
 [-10, 21, 1, -13, -2],  
 [-15, -20, 14, 21, -12],  
 [21, -16, -18, 19, 13],  
 [-13, 5, 19, 23, 9]  
 ]  
  
 vectorB = [16, 9, 2, -3, 16]  
  
 move\_rows\_if\_diagonal\_zero(matrixA, vectorB)  
 lineMethod = "=" \* 160  
 print(lineMethod)  
 match key:  
 case 1:  
 print("Метод Гауса")  
 gauss(matrixA, vectorB)  
 case 2:  
 print("Метод Гауса-Жордана")  
 gauss\_jordan\_with\_steps(matrixA, vectorB)  
 case \_:  
 raise Exception("Погано вибраний метод в use\_method")  
 print(lineMethod)  
  
*# Програма виконання.*if \_\_name\_\_ == "\_\_main\_\_":

print("Програму розробив студент групи ОІ-11 сп, Вальчевський П. В. для варіанту № 3 (згідно списку в журналі) ЛР № 3 з Чисельних методів")  
 use\_method(1)  
 use\_method(2)

**Скріншот отриманого результату.**

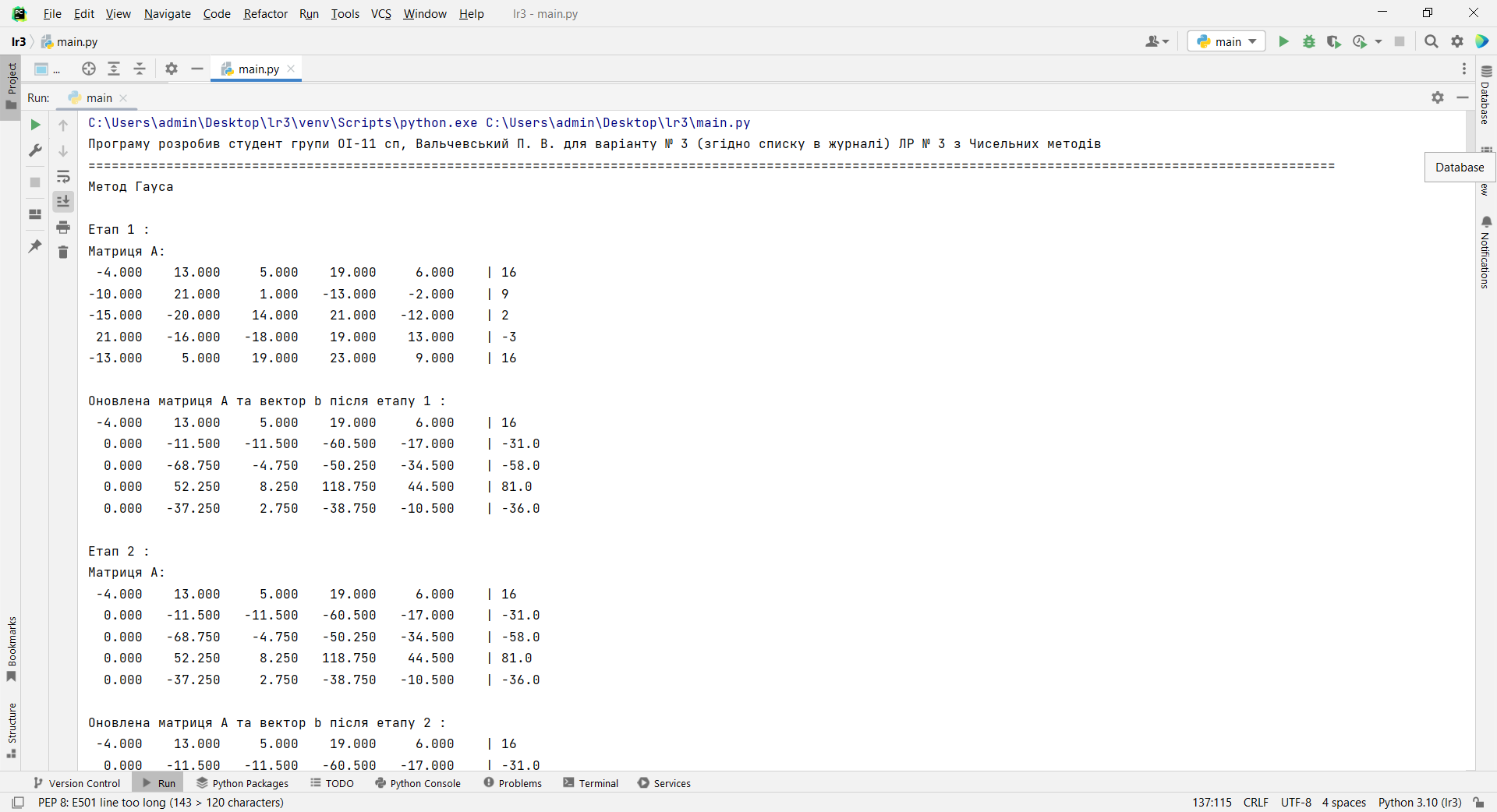


Рис. 1 Результат виконання.

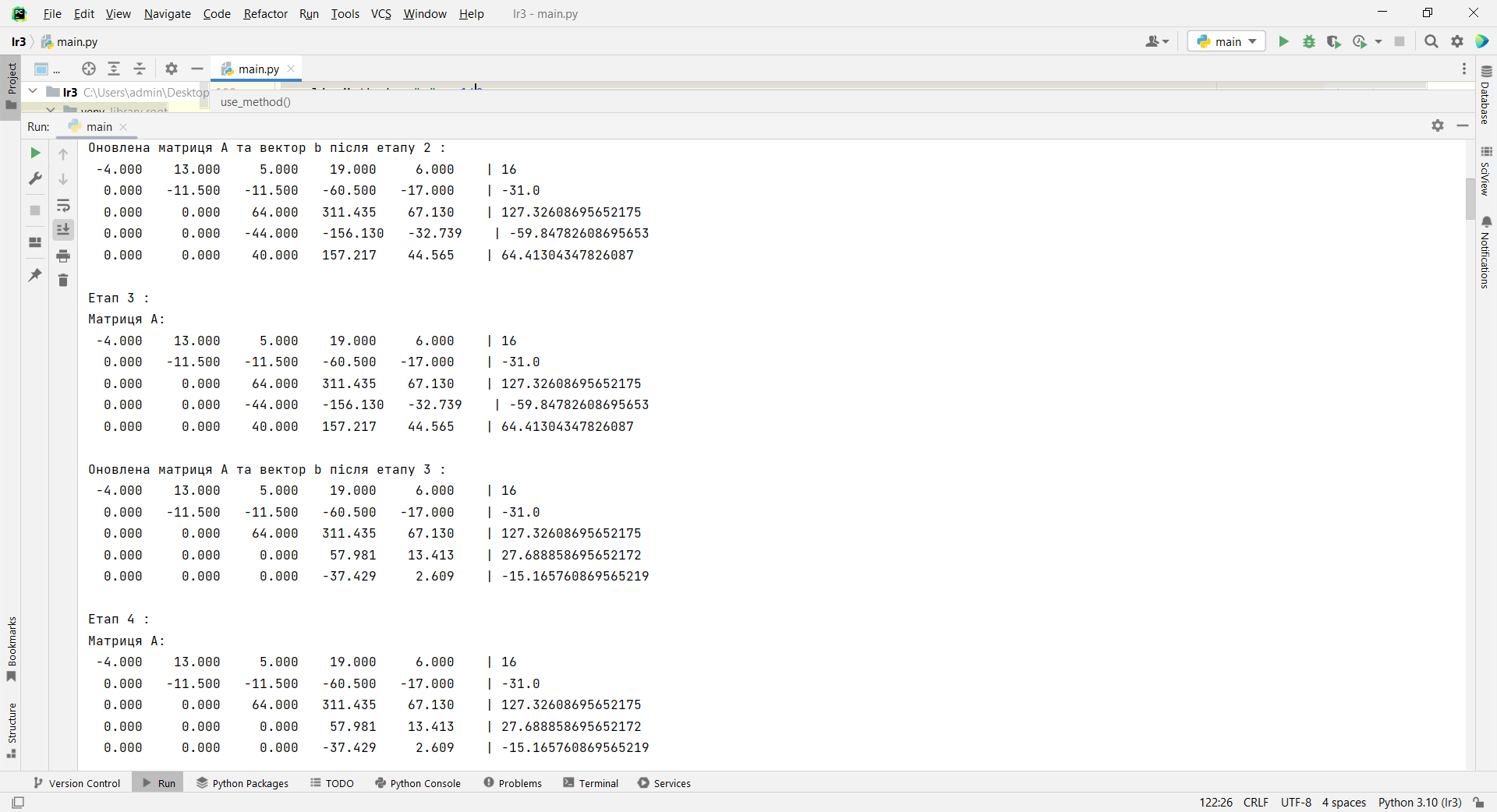


Рис. 2 Результат виконання (продовження).

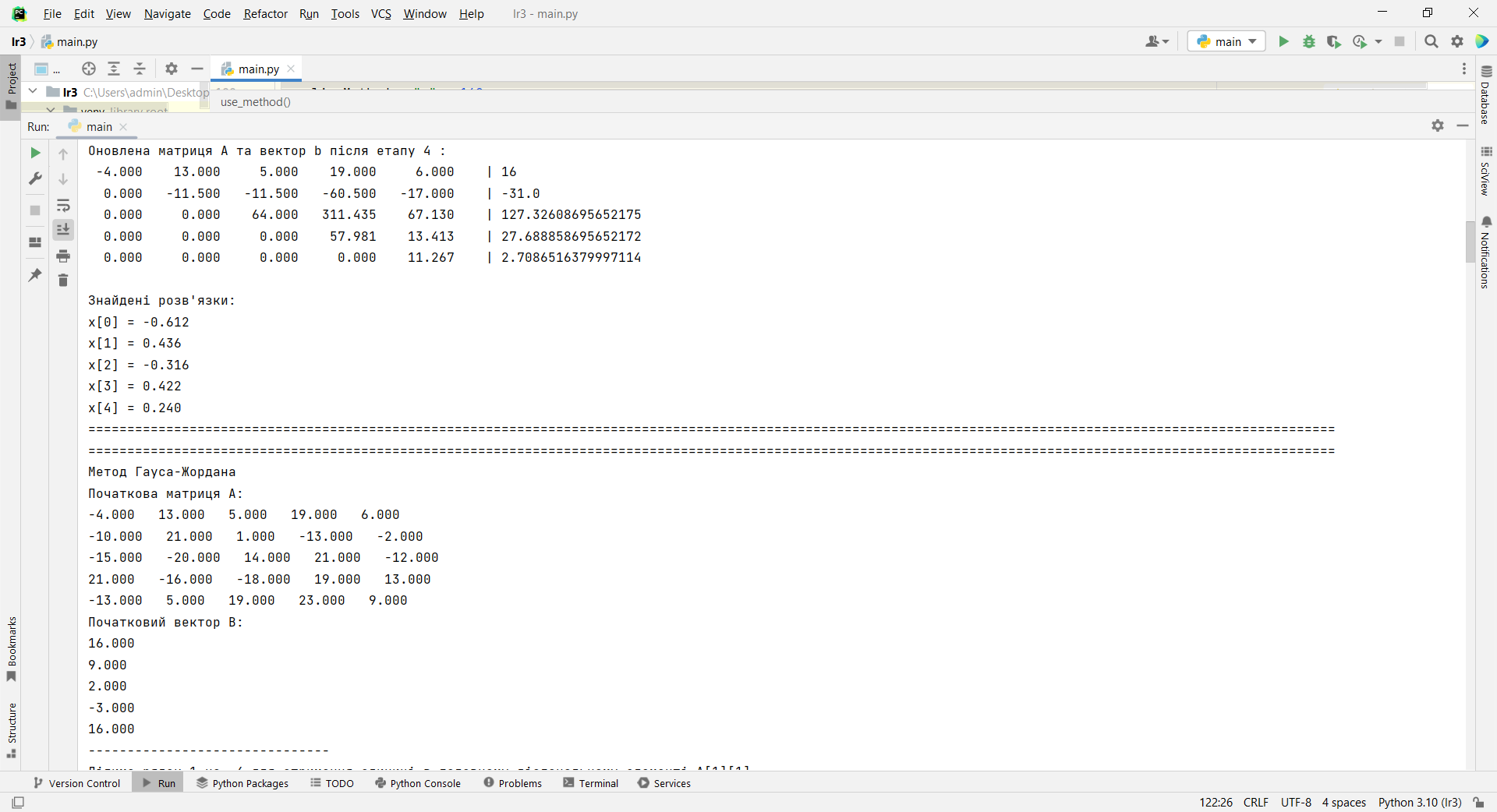


Рис. 3 Результат виконання (продовження).

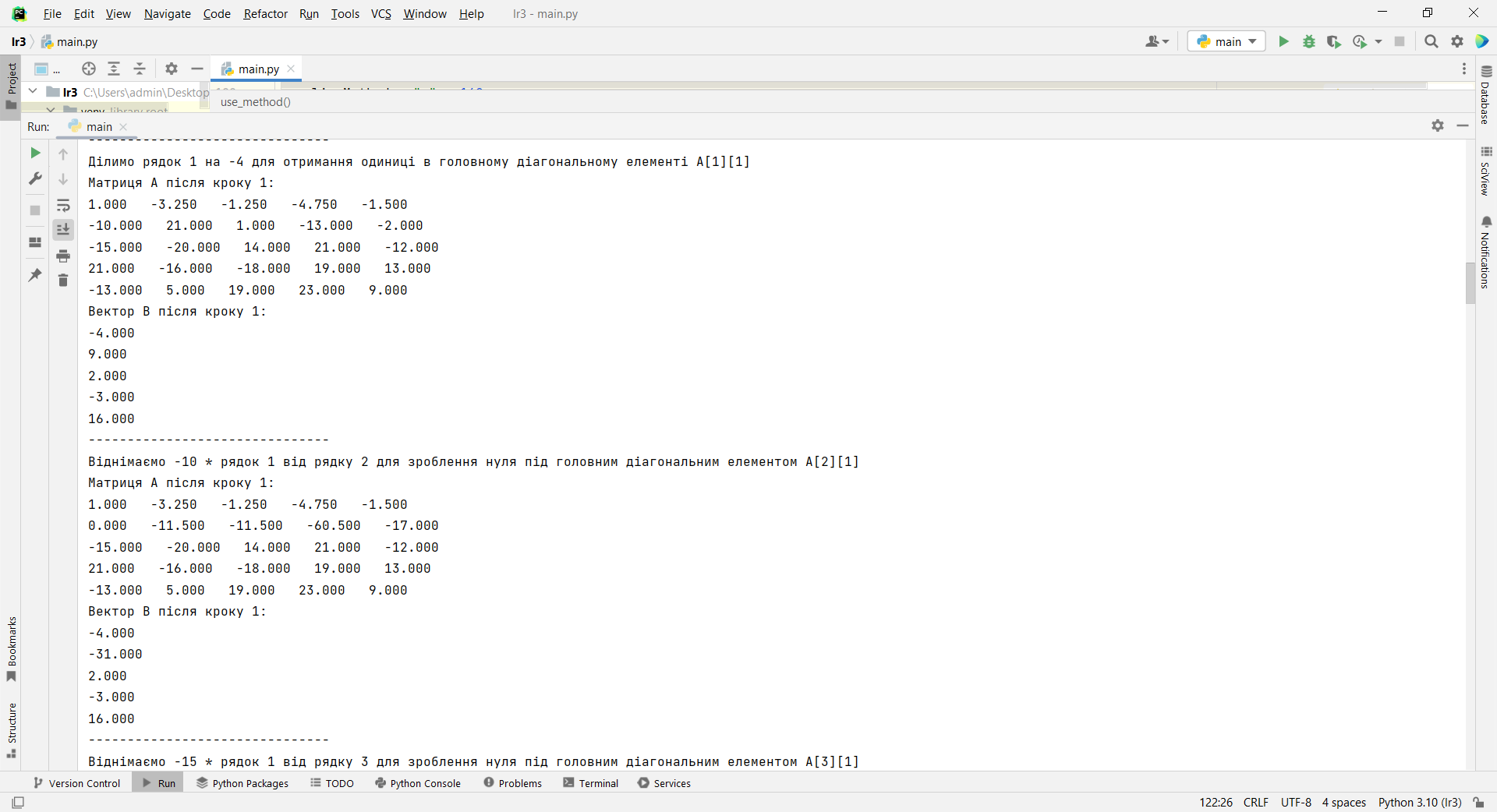


Рис. 4 Результат виконання (продовження).

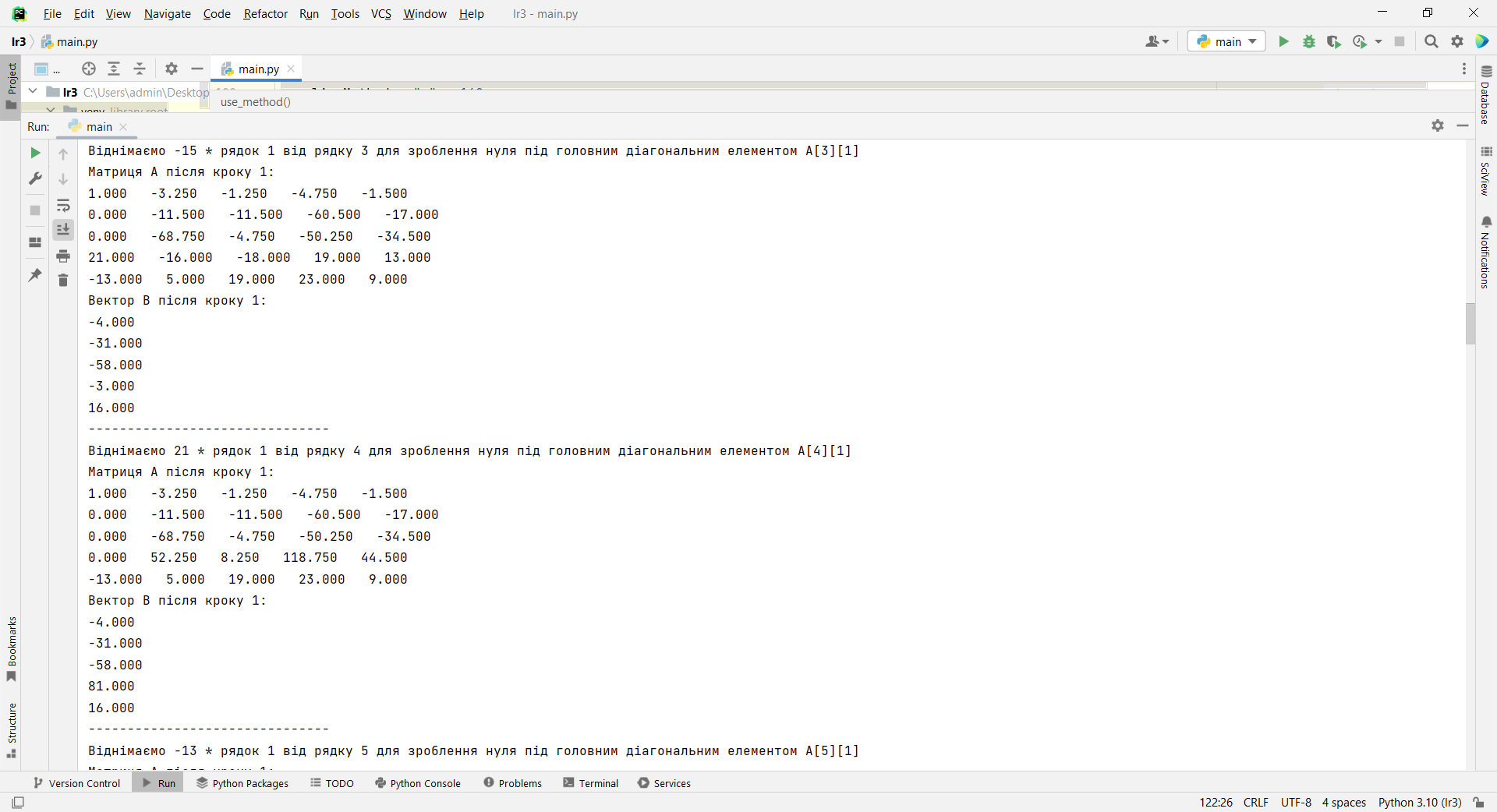


Рис. 5 Результат виконання (продовження).

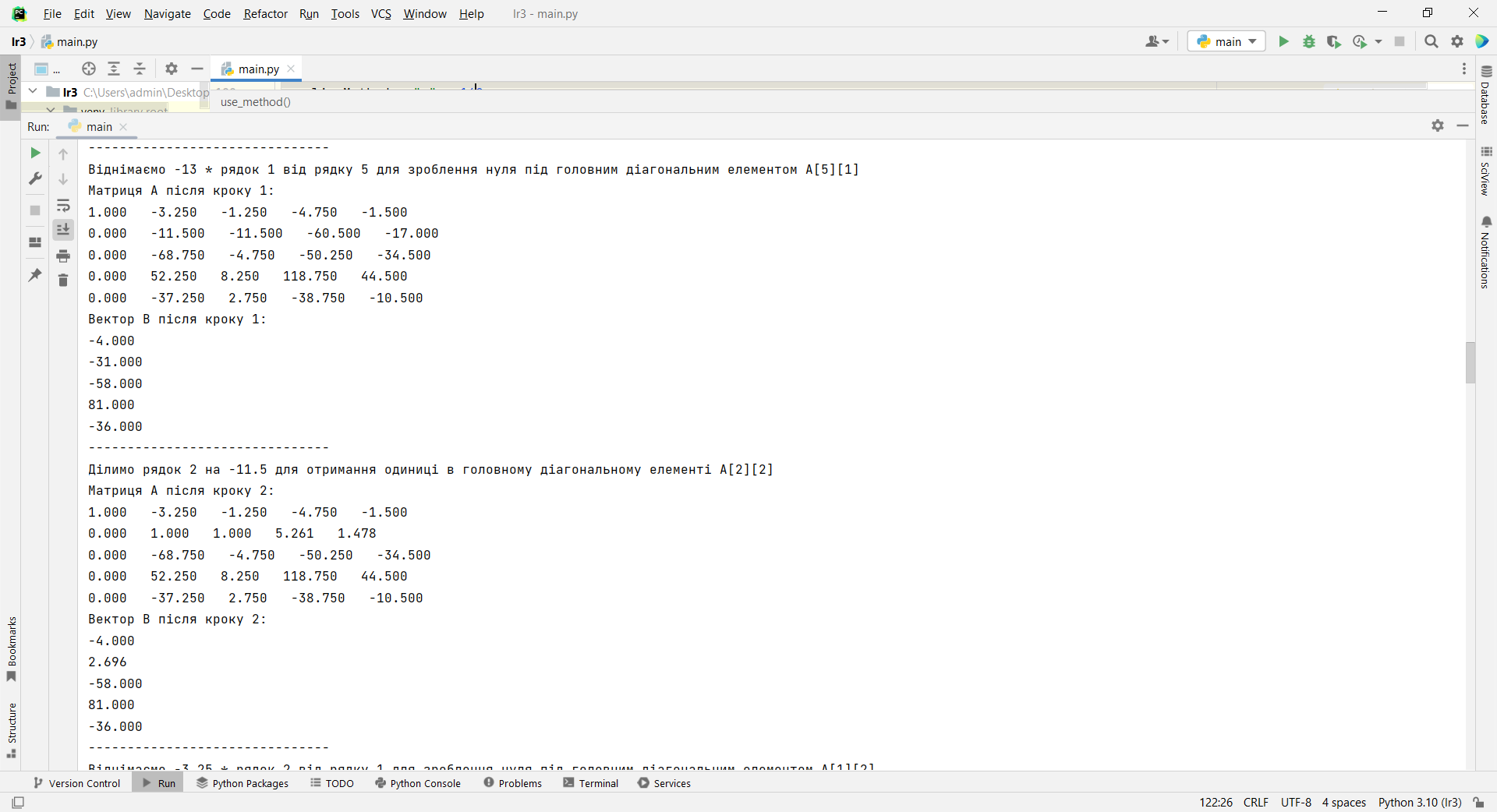


Рис. 6 Результат виконання (продовження).

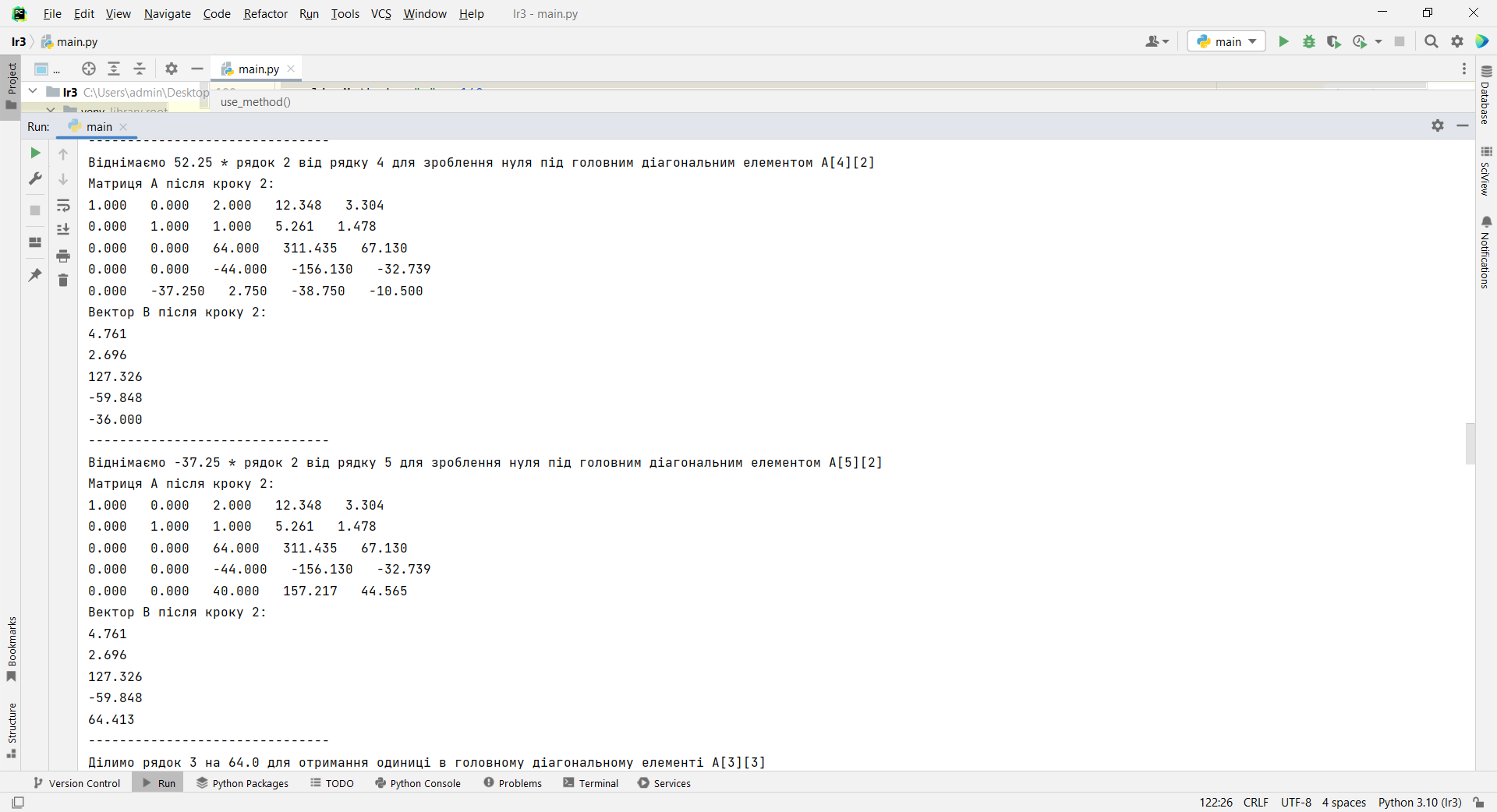


Рис. 7 Результат виконання (продовження).

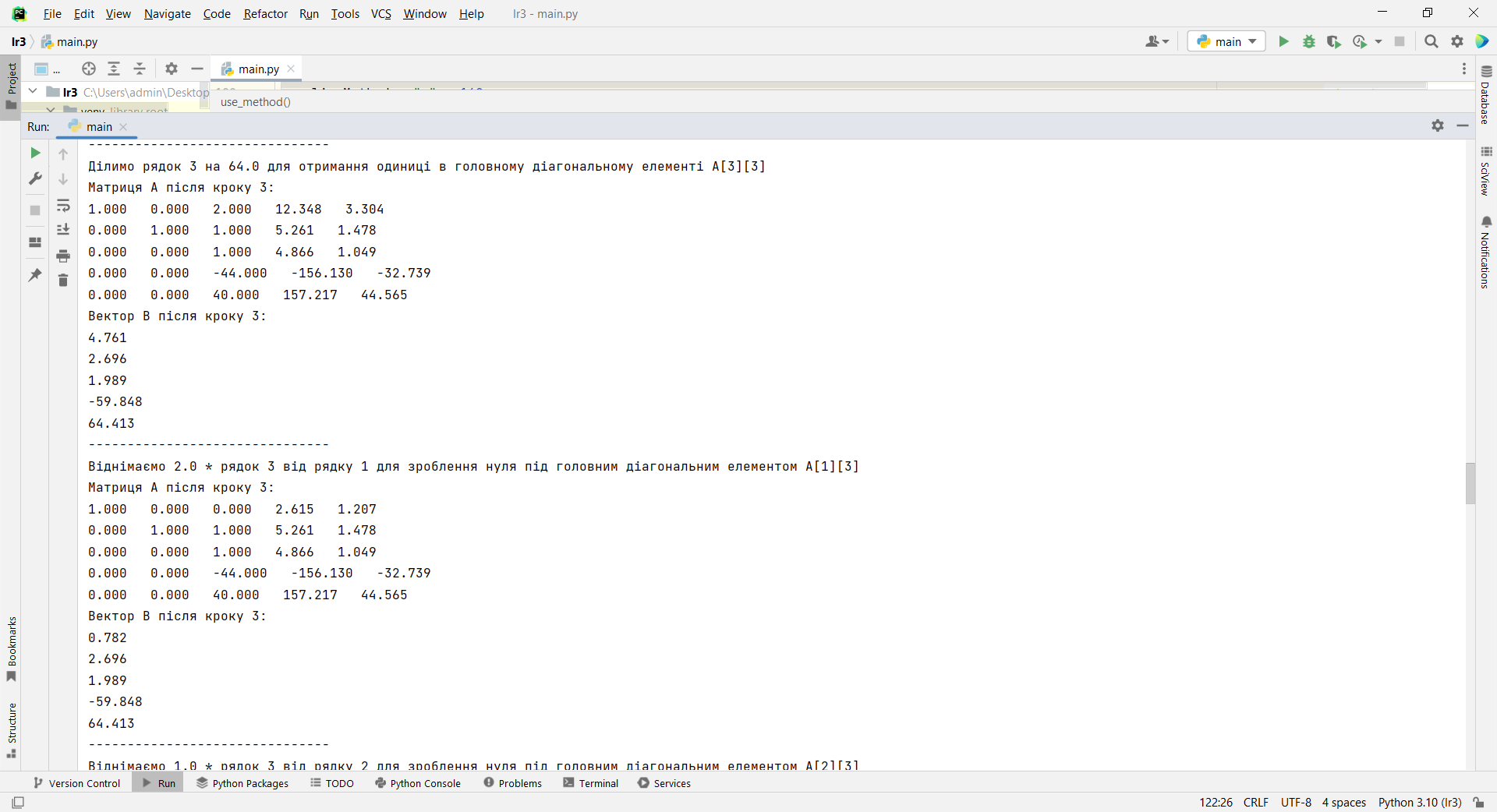


Рис. 8 Результат виконання (продовження).

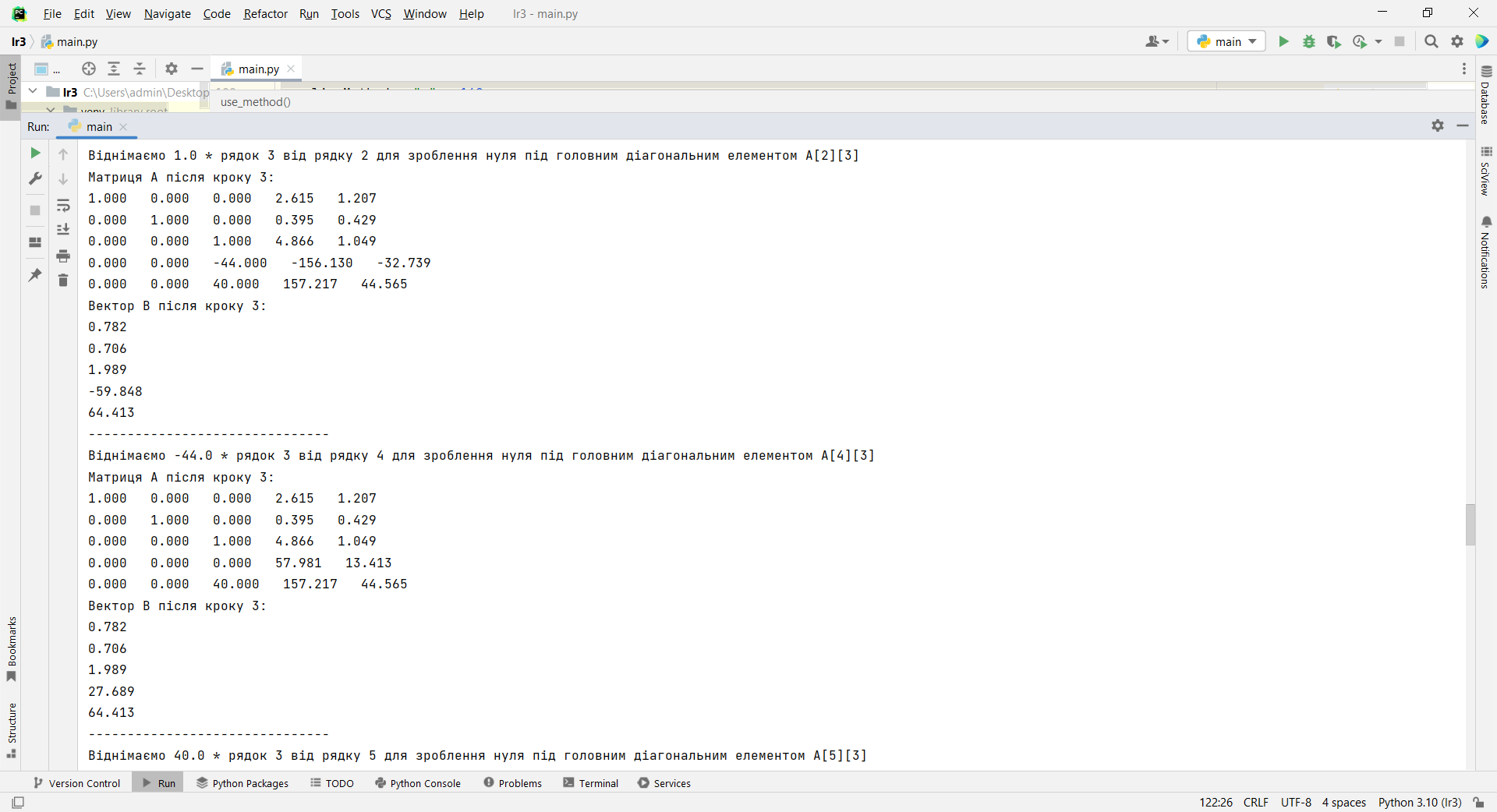


Рис. 9 Результат виконання (продовження).

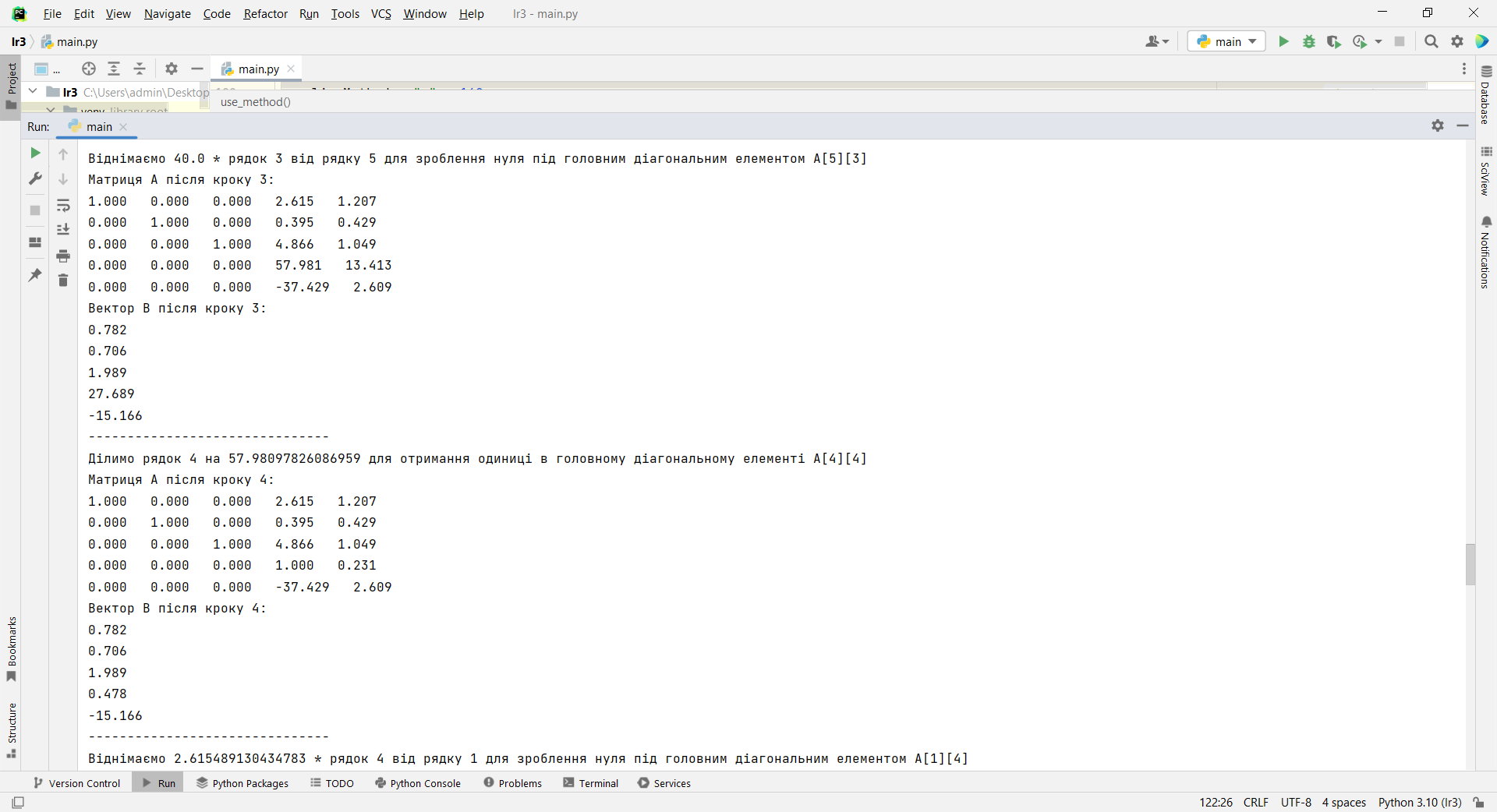


Рис. 10 Результат виконання (продовження).

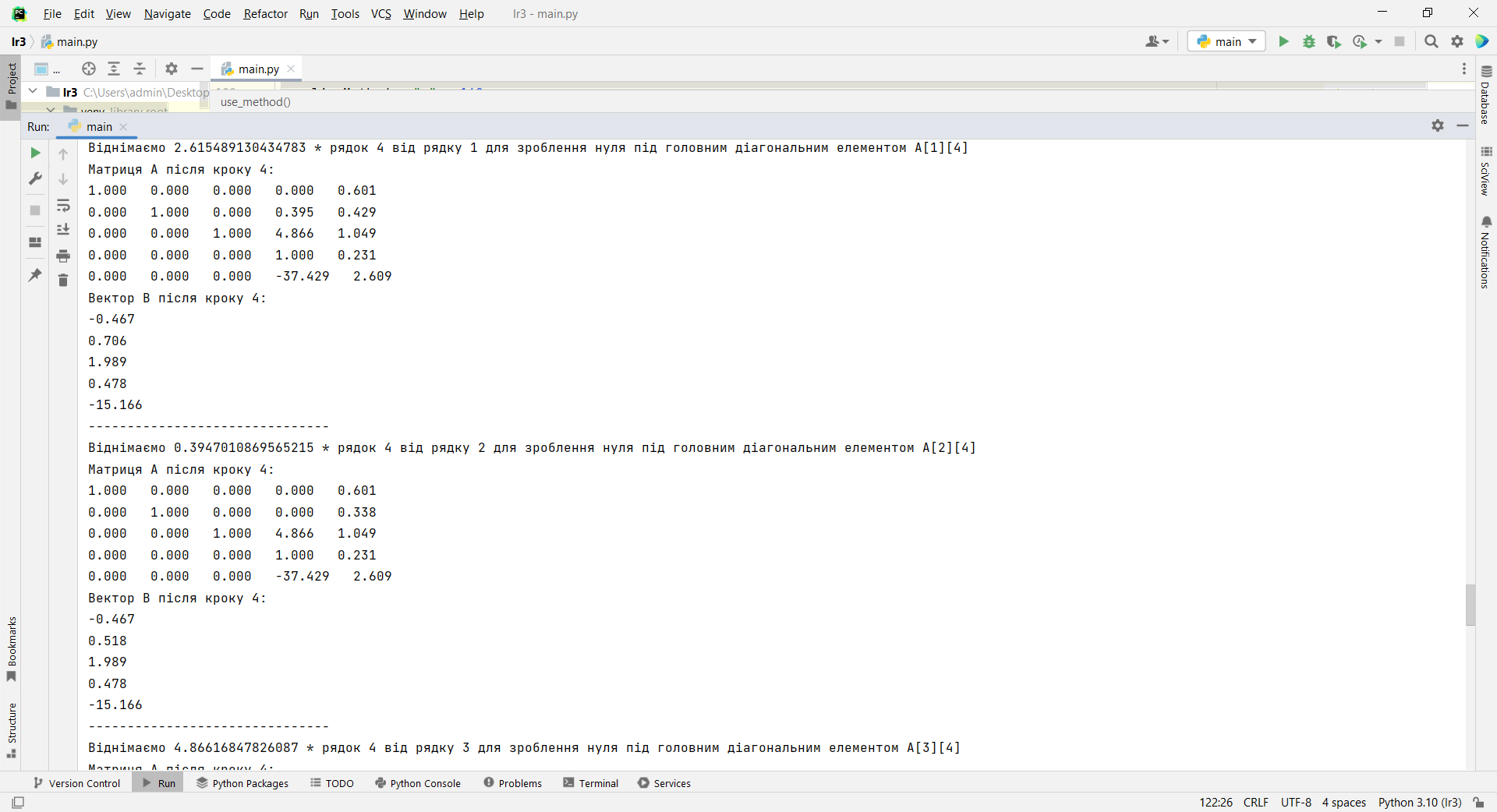


Рис. 11 Результат виконання (продовження).

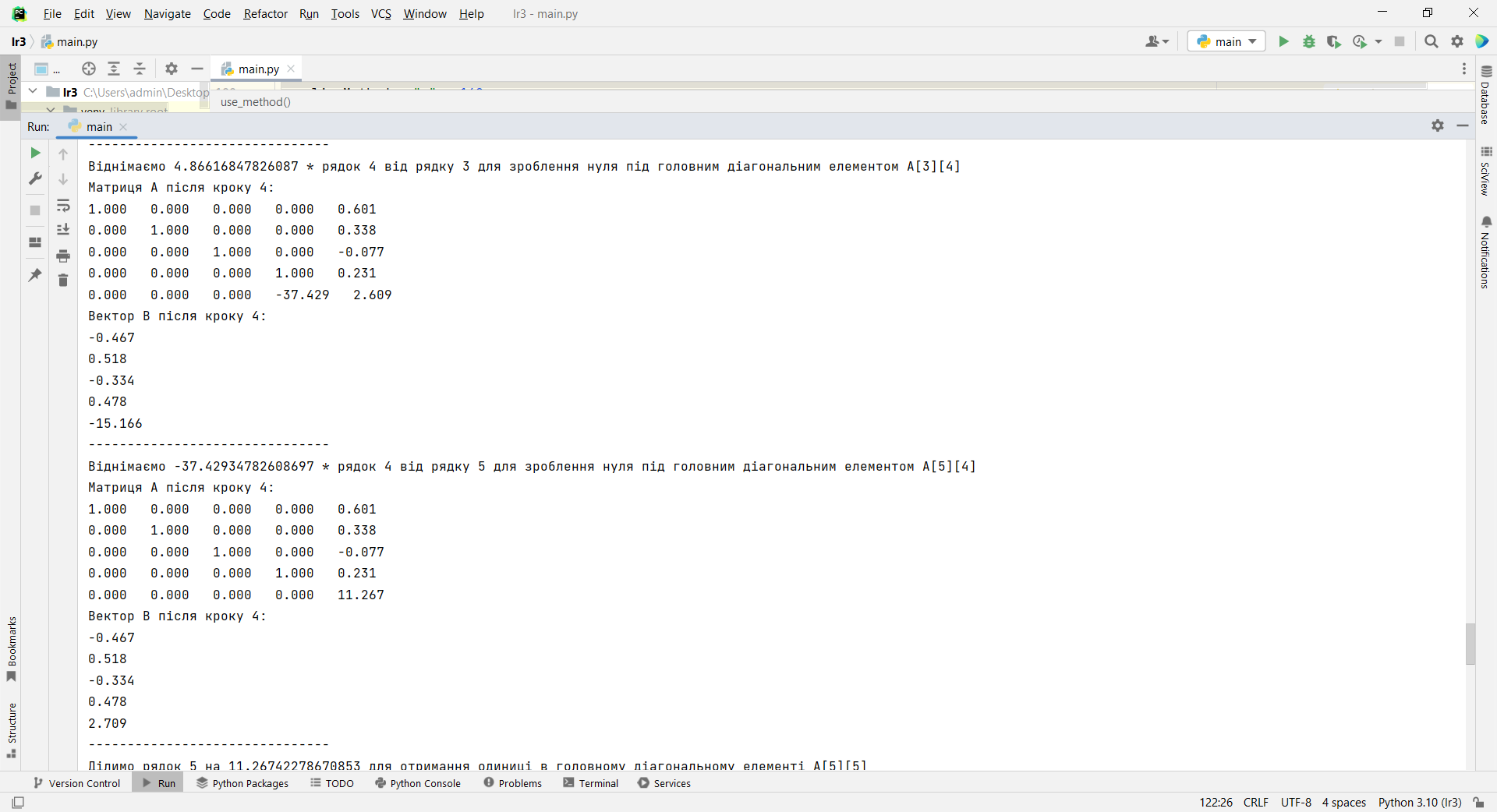


Рис. 12 Результат виконання (продовження).

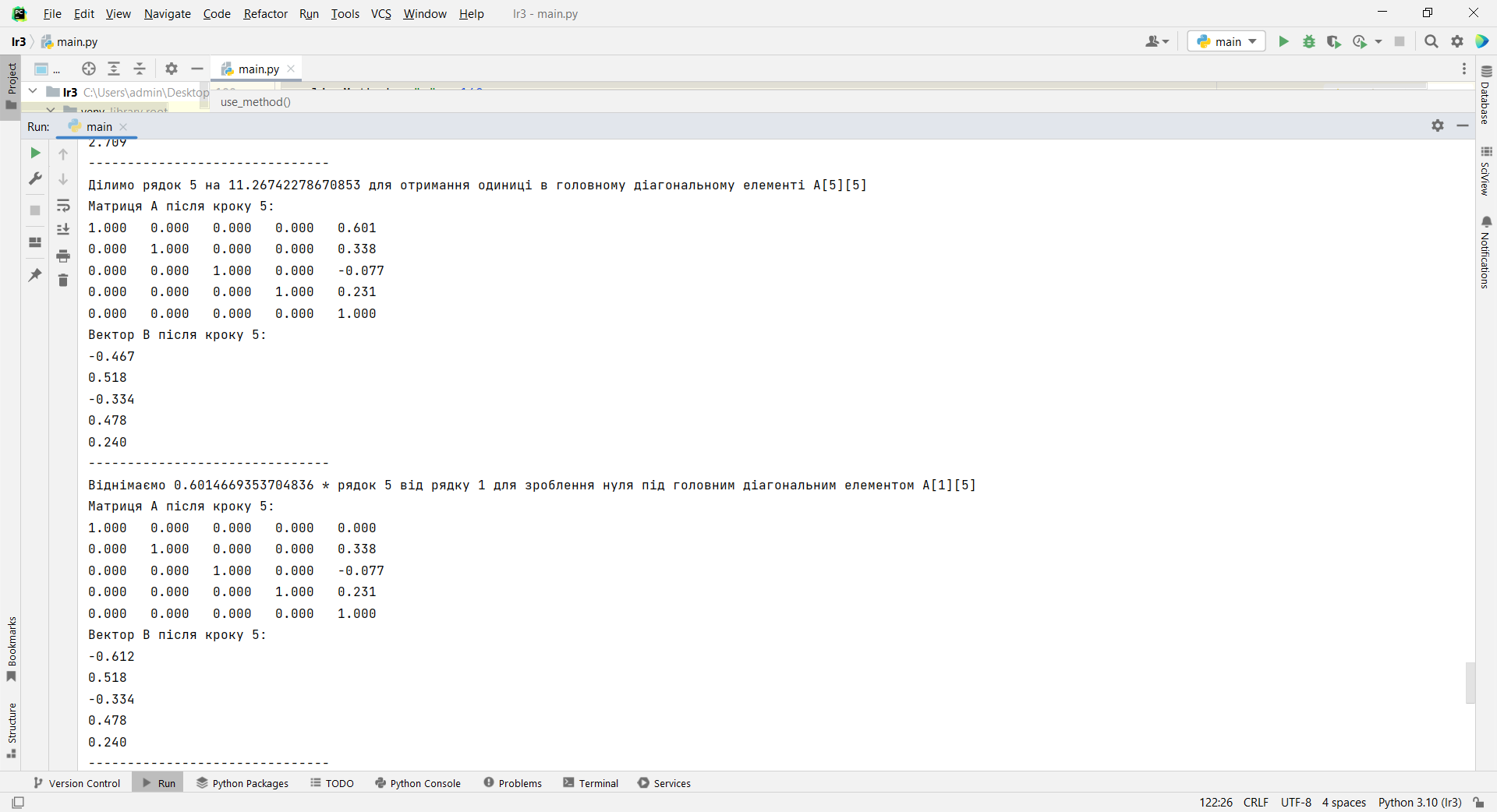


Рис. 13 Результат виконання (продовження).

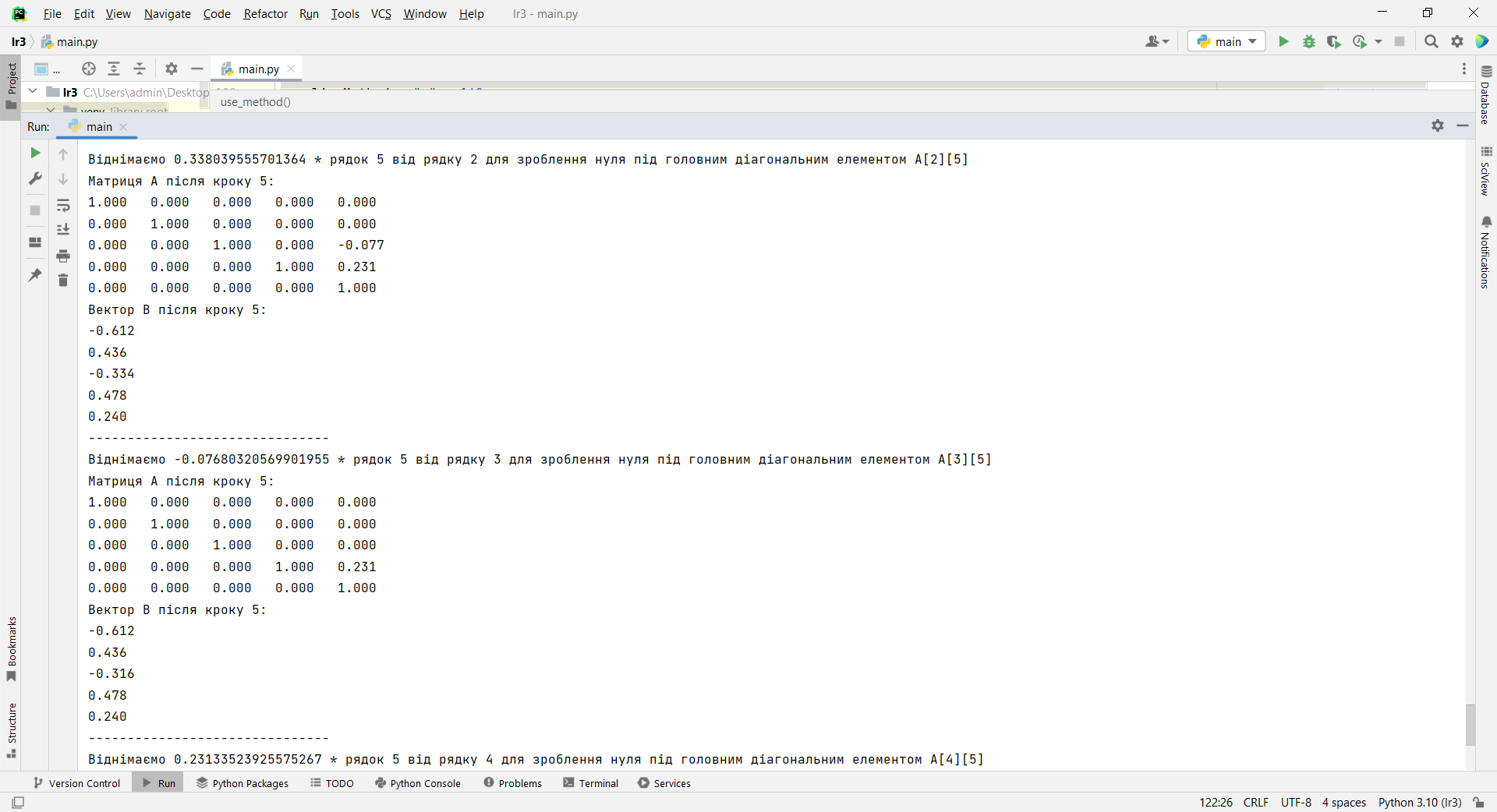


Рис. 14 Результат виконання (продовження).

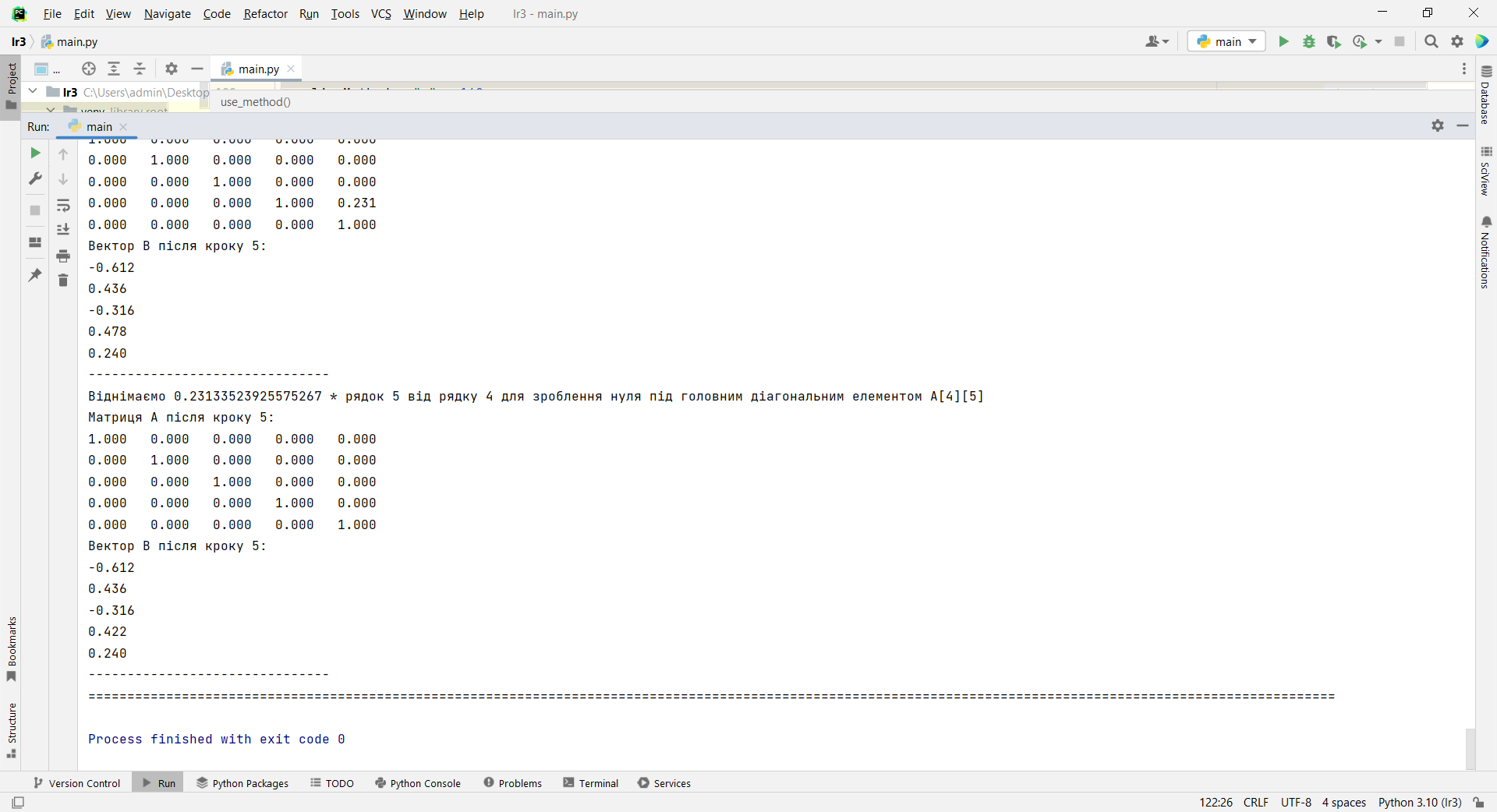


Рис. 15 Результат виконання (продовження).

**Висновки (з аналізом результату).**

У даній лабораторній роботі я ознайомився з методом Гауса та методом Гауса-Жордана. Я розв’язав вручну за допомогою метода Гауса систему рівнянь для подальшої реалізації та перевірки отриманих значень у консолі після виконання програми.

Було реалізовано функції для кожного методу та допоміжні функції для виводу матриці та вектора. Також, було розроблена функція для вибору конкретного методу по ключу (1 – Метод Гауса, 2 – Метод Гауса-Жордана, інше - виключення).

Згідно порівнянь результатів – програма правильно обчислює розв’язок системи рівнянь.

Програму була розроблена в програмному середовищі PyCharm 2023.2.1 мовою програмування Python версії 3.10.4.