**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**



**Лабораторна робота № 1**

з дисципліни «Чисельні методи»

**Виконав:**

Студент групи ШІ-24

Лещишин Маркіян

**Викладач:**

Гентош Л. І.

Львів – 2023р.

**Лабораторна робота № 1**

**Тема:**Метод Гауса для розв’язування систем лінійних алгебричних рівнянь.

**Мета**– засвоїти основні способи практичного використання методу Гауса.

**Варіант: 19**

**Завдання 1.2*.*** Складіть програму, яка знаходить обернену матрицю до

заданої з використанням методу Гауса з постовпцевим вибором головного

елемента.

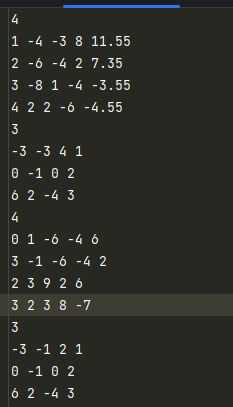
**Хід роботи:**

**Код програми:**

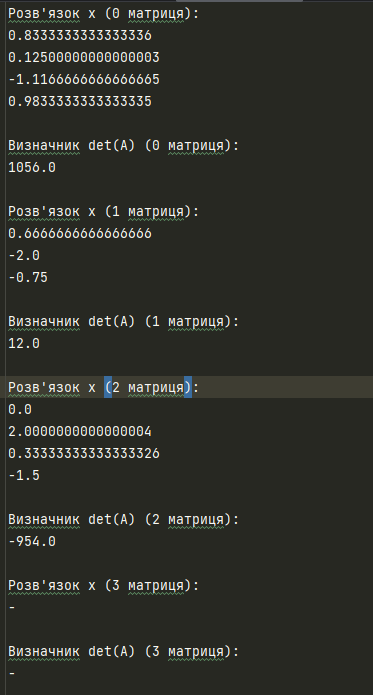
*def* determinant(*matrix*):  
 # Перевірка на те, що матриця є квадратною  
 rows, cols = len(*matrix*), len(*matrix*[0])  
 *if* rows != cols:  
 *raise* ValueError("Матриця не є квадратною")  
  
 # Рекурсивна функція для обчислення визначника  
 *def* recursive\_det(*matrix*):  
 n = len(*matrix*)  
 *if* n == 1:  
 *return matrix*[0][0]  
  
 det = 0  
 *for* j *in* range(n):  
 submatrix = [row[:j] + row[j + 1:] *for* row *in matrix*[1:]]  
 sign = (-1) \*\* j  
 det += sign \* *matrix*[0][j] \* recursive\_det(submatrix)  
  
 *return* det  
  
 *return* recursive\_det(*matrix*)  
  
*def* swap\_rows(*matrix*, *i*, *j*):  
 *matrix*[*i*], *matrix*[*j*] = *matrix*[*j*], *matrix*[*i*]  
  
  
*def* print\_matrix(*matrix*):  
 *for* row *in matrix*:  
 print(row)  
  
  
*def* gaussian\_elimination(*A*, *b*):  
 n = len(*b*)  
 det\_A = 1.0  
  
 # Створення розширеної матриці [A | b]  
 augmented\_matrix = [row + [bi] *for* row, bi *in* zip(*A*, *b*)]  
  
 *for* i *in* range(n):  
 # Пошук максимального елемента у стовпці під поточним  
 max\_row = max(range(i, n), key=*lambda j*: abs(augmented\_matrix[j][i]))  
  
 # Перестановка рядків, якщо потрібно  
 *if* max\_row != i:  
 swap\_rows(augmented\_matrix, i, max\_row)  
 det\_A \*= -1 # Змінюємо знак визначника при обміні рядків  
  
 # Перевірка на нульовий дільник  
 *if* augmented\_matrix[i][i] == 0.0:  
 *raise* ValueError("Матриця не є оберненою або СЛАР не має розв'язку")  
  
 # Елімінація  
 print\_matrix(augmented\_matrix)  
 pivot = augmented\_matrix[i][i]  
 print()  
 *for* j *in* range(i + 1, n):  
 factor = augmented\_matrix[j][i] / pivot  
 print(f"Кожен елемент {j} рядка віднімаємо множник {factor} помноженого на , (елемент {augmented\_matrix[j][i]} ділимо {pivot})")  
 *for* k *in* range(i, n + 1):  
 augmented\_matrix[j][k] -= factor \* augmented\_matrix[i][k]  
 print(f"На {augmented\_matrix[i][k]} , ({i, k})")  
 print\_matrix(augmented\_matrix)  
 print()  
 # Обернена підстановка для знаходження розв'язку x  
  
 x = [0] \* n  
 *for* i *in* range(n - 1, -1, -1):  
 x[i] = augmented\_matrix[i][-1]  
 *for* j *in* range(i + 1, n):  
 x[i] -= augmented\_matrix[i][j] \* x[j]  
 print\_matrix(augmented\_matrix)  
 x[i] /= augmented\_matrix[i][i]  
 print\_matrix(augmented\_matrix)  
 print()  
  
 *return* x, det\_A  
  
  
*def* main():  
 # Зчитування даних з файлу input.txt  
 *with* open("input.txt", "r") *as* file:  
 lines = file.readlines()  
 n = int(lines[0])  
 A = []  
 b = []  
 *for* line *in* lines[1:]:  
 row = [float(x) *for* x *in* line.strip().split()]  
 A.append(row[:-1])  
 b.append(row[-1])  
  
 *try*:  
 # Виклик функції для розв'язання СЛАР  
 x, det\_A = gaussian\_elimination(A, b)  
 det\_A = determinant(A)  
 # Запис результатів у файл output.txt  
 *with* open("output.txt", "w") *as* file:  
 file.write("Розв'язок x:\n")  
 file.write("\n".join(map(str, x)))  
 file.write("\n\nВизначник det(A):\n")  
 file.write(str(det\_A))  
 *except* ValueError *as* e:  
 # Обробка помилки, якщо матриця не є оберненою або СЛАР не має розв'язку  
 print("Помилка:", e)  
  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Результати виконання програми:**

1) input.txt



output.txt



**Висновок:**

На цій лабораторній роботі, я засвоїв основні способи практичного використання методу Гауса.

Загалом, метод Гауса є важливим інструментом для чисельного розв'язування лінійних систем і знаходження розв'язків в різних галузях науки та техніки. Використання цього методу вимагає ретельного підходу до обробки даних і може сприяти ефективному проектуванню математичної програми.