Звіт

# Виконання лабораторної роботи №5

Студента 2ПІ-25Б Заболотного Олександра Івановича

«Робота з багатовимірними масивами»

Тема: Робота з багатовимірними масивами

Мета: Ініціалізація багатовимірних масивів, виконання операцій з масивами.

Завдання 8. Напишіть програму яка отримує від користувача розміри квадратної матриці (n). Потім генерує матрицю А порядку n і вектор b   
з n елементами. Одержати вектор: (A-E) × b, де Е- одинична матриця порядку n.

Текст програми:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX\_N 20

int main() {

    int n;

    float A[MAX\_N][MAX\_N], E[MAX\_N][MAX\_N], C[MAX\_N][MAX\_N], b[MAX\_N], result[MAX\_N];

    printf("Enter max size of square matrix (1-%d): ", MAX\_N);

    scanf("%d", &n);

    if (n < 1 || n > MAX\_N) {

        printf("Invalid size. Please enter a number between 1 and %d.\n", MAX\_N);

        return 1;

    }

    srand(time(NULL));

    // Filling matrix A

    printf("Matrix A:\n");

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            A[i][j] = rand() % 10;

            printf("%5.1f ", A[i][j]);

        }

        printf("\n");

    }

    // Filling vector b

    printf("Vector b:\n");

    for (int i = 0; i < n; i++) {

        b[i] = rand() % 10;

        printf("%5.1f ", b[i]);

    }

    printf("\n");

    // Identity matrix E

    for (int i = 0; i < n; i++)

        for (int j = 0; j < n; j++)

            E[i][j] = (i == j) ? 1 : 0;

    // C = A - E

    for (int i = 0; i < n; i++)

        for (int j = 0; j < n; j++)

            C[i][j] = A[i][j] - E[i][j];

    // result = C \* b

    for (int i = 0; i < n; i++) {

        result[i] = 0;

        for (int j = 0; j < n; j++)

            result[i] += C[i][j] \* b[j];

    }

    printf("Result (A-E)\*b:\n");

    for (int i = 0; i < n; i++)

        printf("%8.2f\n", result[i]);

    return 0;

}

1. Призначення програми:

Програма призначена для генерації квадратної матриці A порядку n, вектора b з n елементами, побудови одиничної матриці E та обчислення вектора (A-E) × b.

2. Вхідні дані:

Вхідні дані — розмір квадратної матриці n (вводиться користувачем).

3. Алгоритм роботи програми:

• Введення розміру матриці n.

• Генерація матриці A розміром n×n з випадкових чисел.

• Генерація вектора b з n випадкових чисел.

• Створення одиничної матриці E порядку n.

• Обчислення матриці C = A - E.

• Множення матриці C на вектор b для отримання результату.

• Виведення матриці A, вектора b та результату (A-E) × b на екран.

4. Опис основних частин коду:

• #include <stdio.h> — підключення бібліотеки для введення/виведення.

• #include <stdlib.h> — підключення стандартної бібліотеки для генерації випадкових чисел.

• #include <time.h> — для ініціалізації генератора випадкових чисел.

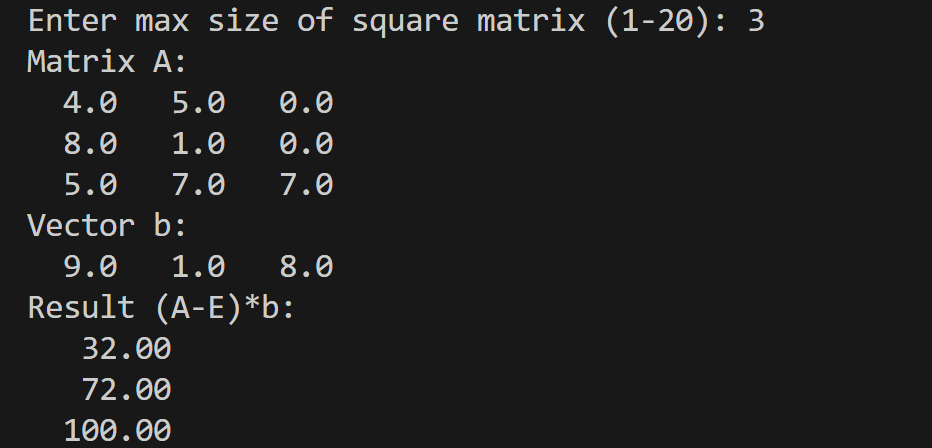
• Оголошення масивів для матриці, вектора, одиничної матриці та результату.

• Введення n через scanf.

• Цикли for для заповнення матриці, вектора, одиничної матриці та обчислення результату.

• Виведення всіх даних на екран.

Результат роботи:



Висновок: У ході виконання лабораторної роботи я навчився оголошувати та використовувати багатовимірні масиви у мові C. Я ознайомився з принципами роботи з матрицями та векторами, навчився заповнювати їх, виконувати арифметичні операції над елементами та реалізовувати алгоритми обробки даних.

В результаті я створив програму, яка генерує квадратну матрицю, вектор, одиничну матрицю та обчислює добуток (A-E) × b згідно з поставленим завданням.

Отримані знання та навички стануть основою для подальшого вивчення програмування та розробки більш складних програм з використанням багатовимірних масивів.  
  
  
Контрольні запитання:  
8. Якого типу масиви є в мові Сі?

За розмірністю:

* Одновимірні масиви: int arr[10];
* Багатовимірні масиви: int matrix[5][3];
* За способом розміщення в пам'яті:
* Статичні масиви (розмір відомий на етапі компіляції): int arr[100];
* Динамічні масиви (розмір визначається під час виконання): через malloc()

За типом елементів:

* Масиви цілих чисел: int arr[10];
* Масиви дійсних чисел: float arr[10];
* Масиви символів (рядки): char str[50];
* Масиви покажчиків: int \*ptrs[10];

9. Як оголосити одновимірний масив використовуючи покажчики?  
Оголошення одновимірного масиву через покажчики:

Статичне виділення пам'яті:

int \*ptr;

int arr[10];

ptr = arr;  // або ptr = &arr[0];

Динамічне виділення пам'яті:

int \*ptr;

int size = 10;

ptr = (int\*)malloc(size \* sizeof(int));

// Не забути звільнити пам'ять: free(ptr);

Доступ до елементів: ptr[i] або \*(ptr + i)

10. Як визначити необхідний розмір пам’яті для розміщення масиву?  
Формула: розмір = кількість\_елементів × sizeof(тип\_елемента)

Щоб отримати розмір масиву, можна скористатися оператором sizeof  
Іншим поширеним способом створення масивів є визначення розміру масиву та додавання елементів пізніше.

Використовуючи цей метод, **ви повинні заздалегідь знати кількість елементів масиву**, щоб програма могла зберігати достатньо пам’яті.