

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки Кафедра
обчислювальної техніки**

Лабораторна робота № 1.5
з дисципліни
«Структури даних і алгоритми»

Виконав: Давидчук А.М.
студент групи ІО-41
Давидчук Артем Миколайович
номер у списку групи: 08

Перевірів:
Сергієнко А. М.

Тема: «Алгоритми лінійного пошуку».

Завдання:

1. Написати програму розв'язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) одним з алгоритмів методу лінійного пошуку.
2. Розміри матриці n та n взяти самостійно у межах від 7 до 10.
3. Виконати тестування та налагодження програми на комп'ютері. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень матриці, щоб можна було легко відстежити коректність виконання пошуку і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

Мій варіант задачі:

Варіант 8

Задано матрицю дійсних чисел $A[n,n]$. У головній діагоналі матриці знайти перший від'ємний і останній додатний елементи, а також поміняти їх місцями.

Обхід матриці реалізував діагональний, алгоритм пошуку значень – лінійний.

Код програми:

```
#include <stdio.h>

int main() {
    int n;

    int negative_coords = 1;
    int positive_coords = -1;

    printf("Enter n number of matrix: ");
    scanf("%d", &n);

    double Matrix[n][n];

    printf("Enter matrix: \n");

    for (int i = 0; i < n; i++){
        for (int j = 0; j < n; j++){
            scanf("%lf", &Matrix[i][j]);
        }
    }

    for (int i = 0; i < n; i++) {
```

```

        if (Matrix[i][i] < 0 && negative_coords == 1) negative_coords = i;
        if (Matrix[i][i] > 0) positive_coords = i;
    }

    if ((negative_coords == 1) || (positive_coords == -1)) {
        printf("\nThere is no negative or positive value in matrix\n");
    }

    else {
        float negative = Matrix[negative_coords][negative_coords];

        Matrix[negative_coords][negative_coords] =
Matrix[positive_coords][positive_coords];
        Matrix[positive_coords][positive_coords] = negative;

        printf("\nResult matrix: \n");

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                printf("%.2lf    ", Matrix[i][j]);
            }

            printf("\n");
        }
    }

    return 0;
}

```

Тестування програм при n=7, n=8, n=10:

```
Enter n number of matrix: 7
Enter matrix:
1.5 -2.1 3.7 -4.4 5.2 6.3 -7.8
-8.6 9.0 -10.4 11.1 -12.8 13.3 14.2
15.9 -16.7 17.5 -18.2 19.4 -20.0 21.1
-22.9 23.6 -24.5 25.4 -26.1 27.8 -28.0
29.3 -30.4 31.6 -32.8 33.2 -34.1 35.5
36.7 -37.9 38.4 -39.2 40.0 41.1 -42.5
-43.6 44.2 -45.1 46.3 -47.8 48.5 -49.9

Result matrix:
1.50 -2.10 3.70 -4.40 5.20 6.30 -7.80
-8.60 9.00 -10.40 11.10 -12.80 13.30 14.20
15.90 -16.70 17.50 -18.20 19.40 -20.00 21.10
-22.90 23.60 -24.50 25.40 -26.10 27.80 -28.00
29.30 -30.40 31.60 -32.80 33.20 -34.10 35.50
36.70 -37.90 38.40 -39.20 40.00 -41.10 -42.50
-43.60 44.20 -45.10 46.30 -47.80 48.50 41.10

Enter n number of matrix: 8
Enter matrix:
-1.2 -2.5 3.1 -4.8 5.4 -6.3 7.0 -8.9
9.1 -10.2 11.5 -12.4 13.6 -14.7 15.0 16.8
-17.3 18.2 -19.5 20.1 -21.6 22.7 23.8 -24.9
25.0 -26.3 27.4 -28.2 29.1 30.5 -31.7 32.9
-33.8 34.1 -35.5 36.2 -37.0 -38.3 39.6 40.2
41.1 -42.9 43.4 -44.0 45.5 -46.1 47.3 48.7
49.0 -50.5 51.2 -52.6 53.7 54.9 -55.8 56.1
57.4 -58.8 59.5 -60.2 61.9 -62.0 63.5 -64.3

There is no negative or positive value in matrix

Enter n number of matrix: 10
Enter matrix:
1.1 -2.3 3.5 -4.7 5.8 -6.1 7.4 -8.2 9.0 -10.6
11.1 -12.5 13.3 -14.6 15.9 -16.8 17.7 -18.0 19.2 20.4
-21.0 22.1 -23.8 24.9 -25.6 26.2 -27.1 28.4 29.9 30.5
31.6 -32.8 33.9 -34.7 35.2 36.5 -37.8 38.0 39.1 40.7
-41.4 42.0 -43.5 44.1 45.6 -46.7 47.3 -48.8 49.0 50.3
51.1 -52.4 53.6 -54.9 55.2 56.0 57.1 -58.5 59.4 60.8
61.7 -62.9 63.1 -64.5 65.3 66.4 -67.6 68.2 69.0 70.5
-71.9 72.3 -73.8 74.1 -75.0 76.6 77.5 -78.3 79.4 80.0
81.2 -82.5 83.1 -84.8 85.0 -86.3 87.7 88.4 -89.2 90.1
-91.0 92.2 -93.3 94.8 -95.5 96.1 -97.0 98.5 99.2 100.0

Result matrix:
1.10 -2.30 3.50 -4.70 5.80 -6.10 7.40 -8.20 9.00 -10.60
11.10 -12.50 13.30 -14.60 15.90 -16.80 17.70 -18.00 19.20 20.40
-21.00 22.10 100.00 24.90 -25.60 26.20 -27.10 28.40 29.90 30.50
31.60 -32.80 33.90 -34.70 35.20 36.50 -37.80 38.00 39.10 40.70
-41.40 42.00 -43.50 44.10 45.60 -46.70 47.30 -48.80 49.00 50.30
51.10 -52.40 53.60 -54.90 55.20 56.00 57.10 -58.50 59.40 60.80
61.70 -62.90 63.10 -64.50 65.30 66.40 -67.60 68.20 69.00 70.50
-71.90 72.30 -73.80 74.10 -75.00 76.60 77.50 -78.30 79.40 80.00
81.20 -82.50 83.10 -84.80 85.00 -86.30 87.70 88.40 -89.20 90.10
-91.00 92.20 -93.30 94.80 -95.50 96.10 -97.00 98.50 99.20 -23.80
```

Висновок:

Я реалізував програму, яка приймає ввід від користувача у вигляді матриці певної розмірності та відповідний вивід. Програма реалізує діагональний обхід на головній діагоналі лінійного пошуку відповідних значень та відповідно до завдання змінює розташування елементів, які підпадають під умову.