

**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  
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-34  
Сюсюков Володимир  
Володимирович  
номер у списку групи: 23

Перевірила:

Молчанова А.А.

Київ 2022

## Завдання

1. Написати програму розв'язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) одним з алгоритмів методу лінійного пошуку.
2. Розміри матриці  $m$  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.

## Варіант 23

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

### Код програми:

```
#include <stdio.h>

#define N 8

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int matrix[N][N];
    printf("Enter the elements of the 8x8 matrix:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
}
```

```

    }
    printf("Initial matrix:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }

    int firstNeg = -1, lastPos = -1;
    for (int i = 0; i < N; i++) {
        if (matrix[i][N - 1 - i] < 0 && firstNeg == -1) {
            firstNeg = i;
        } else if (matrix[i][N - 1 - i] > 0) {
            lastPos = i;
        }
    }

    if (firstNeg != -1 && lastPos != -1) {
        swap(&matrix[firstNeg][N - 1 - firstNeg], &matrix[lastPos][N - 1 - lastPos]);
    }

    printf("\nMatrix after swap:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

## Тестування програми:

Initial matrix:

5	-3	12	-7	8	-11	4	-1
-6	9	-2	10	-13	7	-16	3
11	-4	6	-8	15	-9	2	-14
-12	14	-5	1	-10	13	-15	16
-1	2	-3	4	-5	6	-7	8
-9	10	-11	12	-13	14	-15	16
3	-2	5	-4	7	-6	9	-8
11	-10	13	-12	15	-14	1	-16

Matrix after swap:

5	-3	12	-7	8	-11	4	11
-6	9	-2	10	-13	7	-16	3
11	-4	6	-8	15	-9	2	-14
-12	14	-5	1	-10	13	-15	16
-1	2	-3	4	-5	6	-7	8
-9	10	-11	12	-13	14	-15	16
3	-2	5	-4	7	-6	9	-8
-1	-10	13	-12	15	-14	1	-16

Process finished with exit code 0

Initial matrix:

-10	25	-35	42	-7	18	-29	11
6	-48	37	-3	14	-19	30	-22
-12	9	-41	47	-5	26	-39	20
36	-45	2	-27	38	-16	5	-31
-8	32	-24	10	-1	21	-47	16
-44	28	-37	43	-15	7	-33	13
34	-26	1	-40	29	-6	45	-17
4	-30	39	-11	23	-34	48	-14

Matrix after swap:

-10	25	-35	42	-7	18	-29	11
6	-48	37	-3	14	-19	30	-22
-12	9	-41	47	-5	26	-39	20
36	-45	2	-27	38	-16	5	-31
-8	32	-24	10	-1	21	-47	16
-44	28	4	43	-15	7	-33	13
34	-26	1	-40	29	-6	45	-17
-37	-30	39	-11	23	-34	48	-14

Process finished with exit code 0

Initial matrix:

7	-13	22	-30	6	-45	18	-3
-8	40	-19	11	-27	35	-10	25
-14	4	-36	15	-20	29	-48	9
32	-24	1	-7	17	-42	38	-11
-5	44	-28	12	-2	49	-33	23
-16	31	-25	5	-9	13	-39	27
-21	47	-17	42	-6	19	-29	37
-12	-46	10	-34	30	-18	50	-40

Matrix after swap:

7	-13	22	-30	6	-45	18	47
-8	40	-19	11	-27	35	-10	25
-14	4	-36	15	-20	29	-48	9
32	-24	1	-7	17	-42	38	-11
-5	44	-28	12	-2	49	-33	23
-16	31	-25	5	-9	13	-39	27
-21	-3	-17	42	-6	19	-29	37
-12	-46	10	-34	30	-18	50	-40

Process finished with exit code 0

Initial matrix:

-9	7	-3	12	-20	6	-15	25
5	-30	18	-10	8	-27	35	-14
-5	21	-11	4	-16	29	-7	13
-25	10	-8	19	-2	31	-23	17
15	-40	27	-12	6	-33	22	-18
9	-28	36	-21	3	-17	11	-24
-6	23	-19	16	-13	30	-4	42
20	-37	14	-1	26	-26	38	-22

Matrix after swap:

-9	7	-3	12	-20	6	-15	25
5	-30	18	-10	8	-27	35	-14
-5	21	-11	4	-16	29	-7	13
-25	10	-8	19	20	31	-23	17
15	-40	27	-12	6	-33	22	-18
9	-28	36	-21	3	-17	11	-24
-6	23	-19	16	-13	30	-4	42
-2	-37	14	-1	26	-26	38	-22

Initial matrix:

-14	30	-9	20	-7	11	-26	5
-3	16	-22	38	-10	27	-18	42
9	-33	25	-15	6	19	-29	13
-5	36	-11	7	-24	40	-16	22
32	-8	17	-31	4	-19	10	-27
-2	12	-40	28	-13	35	-6	21
-36	3	-25	14	-1	18	-30	45
23	-4	34	-17	49	-12	2	-38

Matrix after swap:

-14	30	-9	20	-7	11	-26	5
-3	16	-22	38	-10	27	23	42
9	-33	25	-15	6	19	-29	13
-5	36	-11	7	-24	40	-16	22
32	-8	17	-31	4	-19	10	-27
-2	12	-40	28	-13	35	-6	21
-36	3	-25	14	-1	18	-30	45
-18	-4	34	-17	49	-12	2	-38

Process finished with exit code 0