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 №3

з дисципліни
«Алгоритми і структури даних»

Виконав:
студент групи ІМ-43
Олексійчук Станіслав Юрійович
номер у списку групи: 23

Перевірила:
Молчанова А. А.

Київ 2024

Постановка задачі

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.

Тексти програм

Для виконування даної лабораторної я написав програму, яку використовував для 5 різних матриць (масивів), доданих до окремого коду. Ось усі приклади програм:

1) 1st_test

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    double matrix[8][8] = { // створення матриці
        {23.56, -54.32, 12.78, -8.41, 44.67, -78.89, 19.45, -65.23},
        {98.12, -33.67, 14.89, -57.22, 89.34, -14.45, -67.89, 11.43},
        {-72.45, 35.78, -49.34, 76.23, -9.87, -53.11, -61.73, 21.56},
        {66.43, -28.94, 85.13, -64.21, 37.45, -47.67, 91.25, -89.56},
        {54.98, -73.11, 33.56, 9.84, -88.67, 71.43, -92.21, 23.14},
        {-31.67, 92.43, 18.56, 88.12, -57.31, 43.65, -69.78, 62.34},
        {49.23, 91.76, 10.34, -75.43, 83.12, -35.43, 77.89, -23.56},
        {64.89, 18.23, -99.34, 29.67, -48.78, 58.12, -27.45, 73.11}
    };
    int i = 0; //створення змінної для індексів
    int j = 0; //створення змінної для індексів
    // вивід початкової матриці
    printf("{\n");
    for (i = 0; i < 8; i++){
        printf("{");
        for (j = 0; j < 8; j++){
            if (j != 7){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
```

```
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 7){
            printf("\n");
        }
        printf("\n");
    }
    printf("}\n");
    j = 0;
    while (j<8){ //пошук першого від'ємного елемента побічної діагоналі
        if (matrix[7-j][j]<0){
            break;
        }
        j++;
    }
    double first_element = matrix[7-j][j]; //присвоєння змінній значення цього елемента
    printf("The first negative element of side diagonal is %.2f.\n", first_element); //вивід елемента
    i = 0;
    while (i<8){ //пошук останнього додатнього елемента побічної діагоналі
        if (matrix[i][7-i]>0) {
            break;
        }
        i++;
    }
    double second_element = matrix[i][7-i]; //присвоєння змінній значення цього елемента
    printf("The last positive element of side diagonal is %.2f.\n", second_element); //вивід елемента
```

```
    // зміна елементів у матриці між собою
    matrix[7-j][j] = second_element;
    matrix[i][7-i] = first_element;
    // вивід зміненої матриці
    printf("{\n");
    for (i = 0; i < 8; i++){
        printf("{");
        for (j = 0; j < 8; j++){
            if (j != 7){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 7){
            printf("\n");
        }
        printf("\n");
    }
    printf("}");
    return 0;
}
```

2) 2nd_test

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    double matrix[7][7] = { // створення матриці
        {34.56, -72.14, 18.23, -93.78, 56.12, -21.45, 77.89},
```

```
    {-65.43, 14.67, -89.56, 23.12, -41.78, 92.34, -8.91},
    {19.45, -58.67, 81.23, -17.56, 63.89, -32.12, 44.76},
    {78.91, -49.23, 27.89, -63.78, 90.34, -10.45, 52.78},
    {-35.67, 61.23, -88.91, 49.78, -24.56, 84.45, -11.32},
    {93.12, -37.89, 15.67, -69.34, 72.23, -50.89, 38.56},
    {-82.45, 47.12, -22.34, 85.78, -60.11, 13.67, -99.89}
};
int i = 0; //створення змінної для індексів
int j = 0; //створення змінної для індексів
// вивід початкової матриці
printf("{\n");
for (i = 0; i < 7; i++){
    printf("{");
    for (j = 0; j < 7; j++){
        if (j != 6){
            printf("%.2f\t", matrix[i][j]);
        }
        else {
            printf("%.2f", matrix[i][j]);
        }
    }
    printf("}");
    if (i != 6){
        printf("\n");
    }
    printf("\n");
}
printf("}\n");
j = 0;
while (j < 7){ //пошук першого від'ємного елемента побічної діагоналі
    if (matrix[6-j][j]<0){
```

```
            break;
        }
        j++;
    }
    double first_element = matrix[6-j][j]; //присвоєння змінній значення цього елемента
    printf("The first negative element of side diagonal is %.2f.\n", first_element); //вивід елемента
    i = 0;
    while (i < 7){ //пошук останнього додатнього елемента побічної діагоналі
        if (matrix[i][6-i]>0) {
            break;
        }
        i++;
    }
    double second_element = matrix[i][6-i]; //присвоєння змінній значення цього елемента
    printf("The last positive element of side diagonal is %.2f.\n", second_element); //вивід елемента
    // зміна елементів у матриці між собою
    matrix[6-j][j] = second_element;
    matrix[i][6-i] = first_element;
    // вивід зміненої матриці
    printf("{\n");
    for (i = 0; i < 7; i++){
        printf("{");
        for (j = 0; j < 7; j++){
            if (j != 6){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
                printf("%.2f", matrix[i][j]);
```

```
            }
        }
        printf("}");
        if (i != 6){
            printf("\n");
        }
        printf("\n");
    }
    printf("}");
    return 0;
}
```

3) 3rd_test

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    double matrix[9][9] = { // створення матриці
        {45.23, -91.34, 23.67, -76.12, 58.89, -33.45, 82.56, -14.78, -
67.34},
        {-28.12, 72.45, -63.89, 49.78, -15.23, 96.34, -40.56, -29.89, -
85.12},
        {53.67, -47.89, 92.12, -19.34, 64.45, -87.23, -36.78, -60.12,
83.45},
        {-17.89, 78.12, -31.23, 89.34, -55.67, -46.78, -92.45, 30.12, -
74.89},
        {61.34, -42.23, 15.78, -87.12, 93.45, -19.78, 54.12, -32.89,
85.67},
        {-29.45, 90.23, -76.34, 48.12, -69.89, 27.56, -52.12, 83.34, -
16.78},
        {72.56, -14.23, 65.12, -88.45, 39.78, -57.89, 91.34, -26.67,
44.12},
        {-13.78, 82.45, -68.12, 97.34, -33.45, 55.67, -41.23, 78.12, -
53.89},
```

```
        {66.34, -37.12, 21.78, -90.45, 83.23, -19.56, 47.89, -61.34,
38.12}
    };
    int i = 0; //створення змінної для індексів
    int j = 0; //створення змінної для індексів
    // вивід початкової матриці
    printf("{\n");
    for (i = 0; i < 9; i++){
        printf("{");
        for (j = 0; j < 9; j++){
            if (j != 8){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 8){
            printf("\n");
        }
        printf("\n");
    }
    printf("}\n");
    j = 0;
    while (j < 9){ //пошук першого від'ємного елемента побічної діагоналі
        if (matrix[8-j][j]<0){
            break;
        }
        j++;
    }
```

```
    double first_element = matrix[8-j][j]; //присвоєння змінній значення цього елемента
    printf("The first negative element of side diagonal is %.2f.\n", first_element); //вивід елемента
    i = 0;
    while (i < 9){ //пошук останнього додатнього елемента побічної діагоналі
        if (matrix[i][8-i]>0) {
            break;
        }
        i++;
    }
    double second_element = matrix[i][8-i]; //присвоєння змінній значення цього елемента
    printf("The last positive element of side diagonal is %.2f.\n", second_element); //вивід елемента
    // зміна елементів у матриці між собою
    matrix[8-j][j] = second_element;
    matrix[i][8-i] = first_element;
    // вивід зміненої матриці
    printf("{\n");
    for (i = 0; i < 9; i++){
        printf("{");
        for (j = 0; j < 9; j++){
            if (j != 8){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 8){
```

```
            printf("\n");
        }
        printf("\n");
    }
    printf("}");
    return 0;
}
```

4) 4th_test

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    double matrix[10][10] = { //створення матриці
        {72.34, -58.12, 91.23, -49.56, 34.78, -83.67, 18.45, -75.12,
56.34, -20.78},
        {-37.89, 84.23, -61.12, 29.78, -94.34, 53.67, -22.45, 67.12, -
89.34, 41.56},
        {19.45, -76.89, 62.34, -33.12, 85.78, -47.56, 92.23, 14.34,
64.45, -57.12},
        {-25.34, 68.45, -91.67, 34.12, -72.89, 46.78, -38.12, 93.56, -
50.23, 17.89},
        {83.23, -19.12, 48.67, -66.34, 59.12, 32.45, 75.89, -22.34,
98.23, -55.67},
        {-41.56, 92.34, -27.89, 65.23, -73.45, 38.12, -49.34, 87.67, -
15.78, 54.12},
        {31.45, -62.34, 99.12, 43.78, 71.23, -18.56, 52.78, -84.12,
66.34, -29.45},
        {-13.67, 77.89, -58.45, 40.12, -90.34, 35.67, -76.23, 19.78, -
53.12, 68.45},
        {54.12, 38.45, 87.23, -25.12, 61.34, -79.67, 93.45, -46.12,
32.78, -70.89},
        {29.34, 55.78, -67.12, 98.23, -17.45, 82.56, -41.12, 74.89, -
12.67, 63.23}
    };
```

```
    int i = 0; //створення змінної для індексів
    int j = 0; //створення змінної для індексів
    // вивід початкової матриці
    printf("{\n");
    for (i = 0; i < 10; i++){
        printf("{");
        for (j = 0; j < 10; j++){
            if (j != 9){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 9){
            printf("\n");
        }
        printf("\n");

    }
    printf("}\n");
    j = 0;
    while (j < 10){ //пошук першого від'ємного елемента побічної
діагоналі
        if (matrix[9-j][j]<0){
            break;
        }
        j++;
    }
    double first_element = matrix[9-j][j]; //присвоєння змінній значення
цього елемента
```

```
    printf("The first negative element of side diagonal is %.2f.\n", first_element); //вивід елемента

    i = 0;

    while (i < 10){ //пошук останнього додатнього елемента побічної діагоналі

        if (matrix[i][9-i]>0) {

            break;

        }

        i++;

    }

    double second_element = matrix[i][9-i]; //присвоєння змінній значення цього елемента

    printf("The last positive element of side diagonal is %.2f.\n", second_element); //вивід елемента

    // зміна елементів у матриці між собою

    matrix[9-j][j] = second_element;

    matrix[i][9-i] = first_element;

    // вивід зміненої матриці

    printf("{\n");

    for (i = 0; i < 10; i++){

        printf("{");

        for (j = 0; j < 10; j++){

            if (j != 9){

                printf("%.2f\t", matrix[i][j]);

            }

            else {

                printf("%.2f", matrix[i][j]);

            }

        }

        printf("}");

        if (i != 9){

            printf("\n");
```

```
        }
        printf("\n");
    }
    printf("}");
    return 0;
}
```

5) 5th_test

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    double matrix[8][8] = { //створення матриці
        {88.45, -56.12, 72.34, -29.78, 41.56, -83.67, 64.12, -37.89},
        {-27.45, 92.78, -64.34, 55.67, -11.23, 79.12, 36.78, 18.34},
        {14.89, -48.56, 37.12, -82.45, 63.78, 95.12, 20.34, -49.67},
        {-19.34, 45.78, -72.89, 84.12, 60.23, 30.67, -77.45, 99.12},
        {76.12, -11.34, 53.89, 88.67, 22.45, -32.78, 90.34, -5.12},
        {-82.12, 19.78, 74.56, 41.23, -17.89, 89.34, -61.12, 34.45},
        {67.89, 93.45, 15.67, -24.12, 54.78, -42.23, 80.12, -39.67},
        {35.67, 62.34, -88.12, 49.78, -7.23, 33.89, -66.34, 91.56}
    };
    int i = 0; //створення змінної для індексів
    int j = 0; //створення змінної для індексів
    // вивід початкової матриці
    printf("{\n");
    for (i = 0; i < 8; i++){
        printf("{");
        for (j = 0; j < 8; j++){
            if (j != 7){
                printf("%.2f\t", matrix[i][j]);
            }
```

```
            else {
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 7){
            printf("\n");
        }
        printf("\n");
    }
    printf("}\n");
    j = 0;
    while (j<8){ //пошук першого від'ємного елемента побічної діагоналі
        if (matrix[7-j][j]<0){
            break;
        }
        j++;
    }
    double first_element = matrix[7-j][j]; //присвоєння змінній значення цього елемента
    printf("The first negative element of side diagonal is %.2f.\n", first_element); //вивід елемента
    i = 0;
    while (i<8){ //пошук останнього додатнього елемента побічної діагоналі
        if (matrix[i][7-i]>0) {
            break;
        }
        i++;
    }
    double second_element = matrix[i][7-i]; //присвоєння змінній значення цього елемента
```

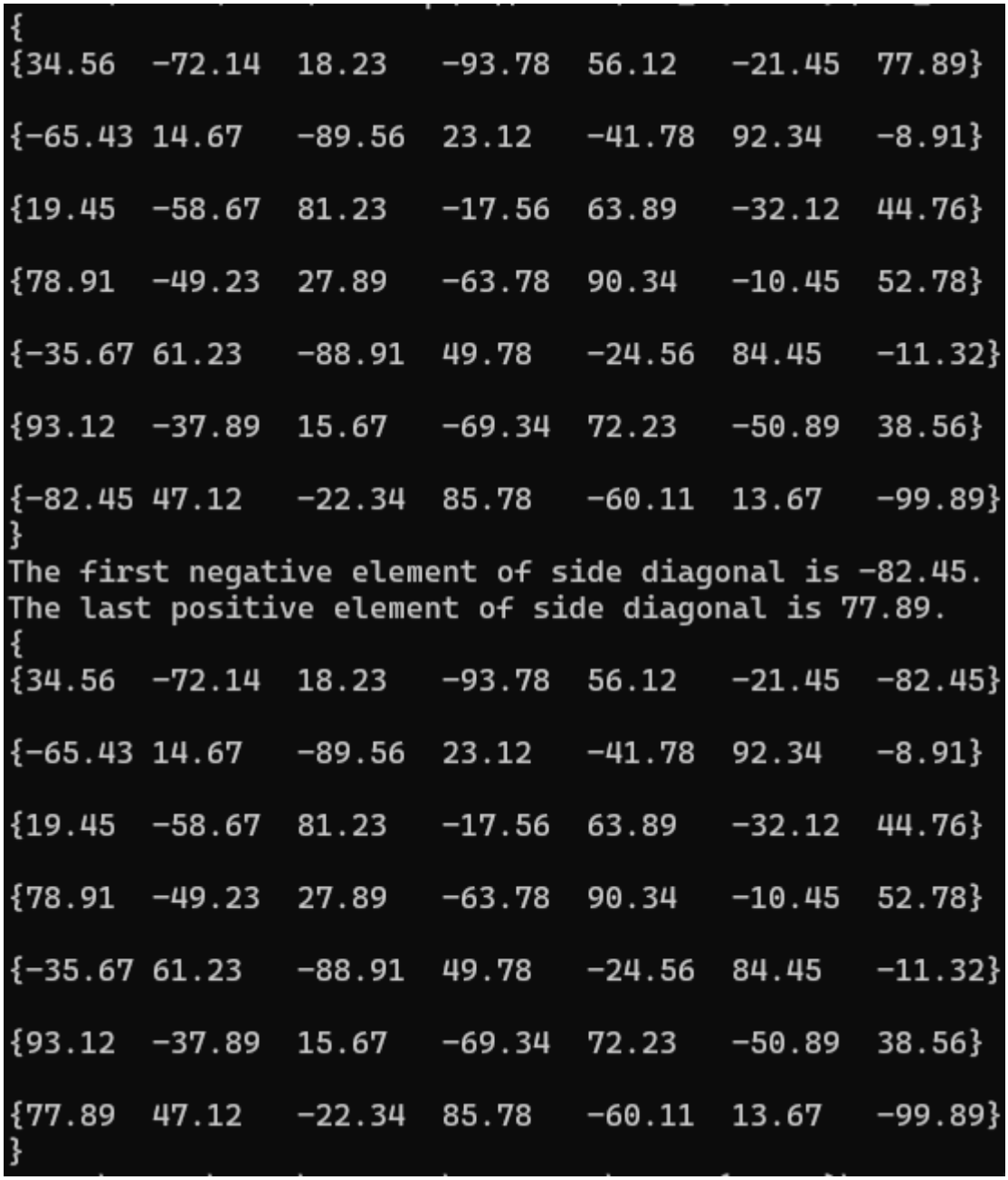
```
    printf("The last positive element of side diagonal is %.2f.\n",
second_element); //вивід елемента
    // зміна елементів у матриці між собою
    matrix[7-j][j] = second_element;
    matrix[i][7-i] = first_element;
    // вивід зміненої матриці
    printf("{\n");
    for (i = 0; i < 8; i++){
        printf("{");
        for (j = 0; j < 8; j++){
            if (j != 7){
                printf("%.2f\t", matrix[i][j]);
            }
            else {
                printf("%.2f", matrix[i][j]);
            }
        }
        printf("}");
        if (i != 7){
            printf("\n");
        }
        printf("\n");
    }
    printf("}");
    return 0;
}
```

Результати тестування програм

1)

```
{
{23.56  -54.32  12.78   -8.41   44.67   -78.89  19.45   -65.23}

{98.12  -33.67  14.89   -57.22  89.34   -14.45  -67.89  11.43}

{-72.45 35.78   -49.34  76.23   -9.87   -53.11  -61.73  21.56}

{66.43  -28.94  85.13   -64.21  37.45   -47.67  91.25   -89.56}

{54.98  -73.11  33.56   9.84    -88.67  71.43   -92.21  23.14}

{-31.67 92.43   18.56   88.12   -57.31  43.65   -69.78  62.34}

{49.23  91.76   10.34   -75.43  83.12   -35.43  77.89   -23.56}

{64.89  18.23   -99.34  29.67   -48.78  58.12   -27.45  73.11}
}
The first negative element of side diagonal is -53.11.
The last positive element of side diagonal is 37.45.
{
{23.56  -54.32  12.78   -8.41   44.67   -78.89  19.45   -65.23}

{98.12  -33.67  14.89   -57.22  89.34   -14.45  -67.89  11.43}

{-72.45 35.78   -49.34  76.23   -9.87   37.45   -61.73  21.56}

{66.43  -28.94  85.13   -64.21  -53.11  -47.67  91.25   -89.56}

{54.98  -73.11  33.56   9.84    -88.67  71.43   -92.21  23.14}

{-31.67 92.43   18.56   88.12   -57.31  43.65   -69.78  62.34}

{49.23  91.76   10.34   -75.43  83.12   -35.43  77.89   -23.56}

{64.89  18.23   -99.34  29.67   -48.78  58.12   -27.45  73.11}
}
```

2)


{
{34.56 -72.14 18.23 -93.78 56.12 -21.45 77.89}
{-65.43 14.67 -89.56 23.12 -41.78 92.34 -8.91}
{19.45 -58.67 81.23 -17.56 63.89 -32.12 44.76}
{78.91 -49.23 27.89 -63.78 90.34 -10.45 52.78}
{-35.67 61.23 -88.91 49.78 -24.56 84.45 -11.32}
{93.12 -37.89 15.67 -69.34 72.23 -50.89 38.56}
{-82.45 47.12 -22.34 85.78 -60.11 13.67 -99.89}
}
The first negative element of side diagonal is -82.45.
The last positive element of side diagonal is 77.89.
{
{34.56 -72.14 18.23 -93.78 56.12 -21.45 -82.45}
{-65.43 14.67 -89.56 23.12 -41.78 92.34 -8.91}
{19.45 -58.67 81.23 -17.56 63.89 -32.12 44.76}
{78.91 -49.23 27.89 -63.78 90.34 -10.45 52.78}
{-35.67 61.23 -88.91 49.78 -24.56 84.45 -11.32}
{93.12 -37.89 15.67 -69.34 72.23 -50.89 38.56}
{77.89 47.12 -22.34 85.78 -60.11 13.67 -99.89}
}

3)

```
{
{45.23  -91.34  23.67   -76.12  58.89   -33.45  82.56   -14.78  -67.34}

{-28.12 72.45   -63.89  49.78   -15.23  96.34   -40.56  -29.89  -85.12}

{53.67  -47.89  92.12   -19.34  64.45   -87.23  -36.78  -60.12  83.45}

{-17.89 78.12   -31.23  89.34   -55.67  -46.78  -92.45  30.12   -74.89}

{61.34  -42.23  15.78   -87.12  93.45   -19.78  54.12   -32.89  85.67}

{-29.45 90.23   -76.34  48.12   -69.89  27.56   -52.12  83.34   -16.78}

{72.56  -14.23  65.12   -88.45  39.78   -57.89  91.34   -26.67  44.12}

{-13.78 82.45   -68.12  97.34   -33.45  55.67   -41.23  78.12   -53.89}

{66.34  -37.12  21.78   -90.45  83.23   -19.56  47.89   -61.34  38.12}
}
The first negative element of side diagonal is -46.78.
The last positive element of side diagonal is 93.45.
{
{45.23  -91.34  23.67   -76.12  58.89   -33.45  82.56   -14.78  -67.34}

{-28.12 72.45   -63.89  49.78   -15.23  96.34   -40.56  -29.89  -85.12}

{53.67  -47.89  92.12   -19.34  64.45   -87.23  -36.78  -60.12  83.45}

{-17.89 78.12   -31.23  89.34   -55.67  93.45   -92.45  30.12   -74.89}

{61.34  -42.23  15.78   -87.12  -46.78  -19.78  54.12   -32.89  85.67}

{-29.45 90.23   -76.34  48.12   -69.89  27.56   -52.12  83.34   -16.78}

{72.56  -14.23  65.12   -88.45  39.78   -57.89  91.34   -26.67  44.12}

{-13.78 82.45   -68.12  97.34   -33.45  55.67   -41.23  78.12   -53.89}

{66.34  -37.12  21.78   -90.45  83.23   -19.56  47.89   -61.34  38.12}
}
```

4)

```
{
{72.34  -58.12  91.23   -49.56  34.78   -83.67  18.45   -75.12  56.34   -20.78}

{-37.89 84.23   -61.12  29.78   -94.34  53.67   -22.45  67.12   -89.34  41.56}

{19.45  -76.89  62.34   -33.12  85.78   -47.56  92.23   14.34   64.45   -57.12}

{-25.34 68.45   -91.67  34.12   -72.89  46.78   -38.12  93.56   -50.23  17.89}

{83.23  -19.12  48.67   -66.34  59.12   32.45   75.89   -22.34  98.23   -55.67}

{-41.56 92.34   -27.89  65.23   -73.45  38.12   -49.34  87.67   -15.78  54.12}

{31.45  -62.34  99.12   43.78   71.23   -18.56  52.78   -84.12  66.34   -29.45}

{-13.67 77.89   -58.45  40.12   -90.34  35.67   -76.23  19.78   -53.12  68.45}

{54.12  38.45   87.23   -25.12  61.34   -79.67  93.45   -46.12  32.78   -70.89}

{29.34  55.78   -67.12  98.23   -17.45  82.56   -41.12  74.89   -12.67  63.23}
}
The first negative element of side diagonal is -58.45.
The last positive element of side diagonal is 14.34.
{
{72.34  -58.12  91.23   -49.56  34.78   -83.67  18.45   -75.12  56.34   -20.78}

{-37.89 84.23   -61.12  29.78   -94.34  53.67   -22.45  67.12   -89.34  41.56}

{19.45  -76.89  62.34   -33.12  85.78   -47.56  92.23   -58.45  64.45   -57.12}

{-25.34 68.45   -91.67  34.12   -72.89  46.78   -38.12  93.56   -50.23  17.89}

{83.23  -19.12  48.67   -66.34  59.12   32.45   75.89   -22.34  98.23   -55.67}

{-41.56 92.34   -27.89  65.23   -73.45  38.12   -49.34  87.67   -15.78  54.12}

{31.45  -62.34  99.12   43.78   71.23   -18.56  52.78   -84.12  66.34   -29.45}

{-13.67 77.89   14.34   40.12   -90.34  35.67   -76.23  19.78   -53.12  68.45}

{54.12  38.45   87.23   -25.12  61.34   -79.67  93.45   -46.12  32.78   -70.89}

{29.34  55.78   -67.12  98.23   -17.45  82.56   -41.12  74.89   -12.67  63.23}
}
```

5)

```
{
{88.45  -56.12  72.34   -29.78  41.56   -83.67  64.12   -37.89}

{-27.45 92.78   -64.34  55.67   -11.23  79.12   36.78   18.34}

{14.89  -48.56  37.12   -82.45  63.78   95.12   20.34   -49.67}

{-19.34 45.78   -72.89  84.12   60.23   30.67   -77.45  99.12}

{76.12  -11.34  53.89   88.67   22.45   -32.78  90.34   -5.12}

{-82.12 19.78   74.56   41.23   -17.89  89.34   -61.12  34.45}

{67.89  93.45   15.67   -24.12  54.78   -42.23  80.12   -39.67}

{35.67  62.34   -88.12  49.78   -7.23   33.89   -66.34  91.56}
}
The first negative element of side diagonal is -37.89.
The last positive element of side diagonal is 36.78.
{
{88.45  -56.12  72.34   -29.78  41.56   -83.67  64.12   36.78}

{-27.45 92.78   -64.34  55.67   -11.23  79.12   -37.89  18.34}

{14.89  -48.56  37.12   -82.45  63.78   95.12   20.34   -49.67}

{-19.34 45.78   -72.89  84.12   60.23   30.67   -77.45  99.12}

{76.12  -11.34  53.89   88.67   22.45   -32.78  90.34   -5.12}

{-82.12 19.78   74.56   41.23   -17.89  89.34   -61.12  34.45}

{67.89  93.45   15.67   -24.12  54.78   -42.23  80.12   -39.67}

{35.67  62.34   -88.12  49.78   -7.23   33.89   -66.34  91.56}
}
```

Висновок: я навчився створювати двовимірні масиви у мові програмування С та далі опрацьовувати їх: знаходити будь-який елемент, знаючи індекси, змінювати місцями елементи масивів і виводити масиви у вигляді матриці; зрозумів, як працюють індекси у двовимірних масивах, як можна задати розмірність масиву, і, слухаючи та аналізуючи лекції до та після написання програм, зрозумів, як комп’ютер розглядає масиви; зміг проходити елементи масиву, використовуючи оператори циклу та розгалуження; зрозумів, як можна задати розмірність масиву.