

**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-42
Туров Андрій Володимирович
номер у списку групи: 29

Перевірила:

Молчанова А. А.

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 29

Задано матрицю дійсних чисел $A[n; n]$. У побічній діагоналі матриці визначити присутність заданого дійсного числа X і його місцезнаходження (координати).

Текст програми

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

int main(void) {
    srand(clock());

    int size;
    printf("Enter the matrix size (n*n): ");
    scanf("%d", &size);
    printf("\n");

    float matrix[size][size];

    for (int y = 0; y < size; y++) {
        for (int x = 0; x < size; x++) {
            matrix[y][x] = ((float) rand() / (float) RAND_MAX) * 99.0;
            /*matrix[y][x] = abs(x - y); // i'm leaving it here for testing multiple
occurences*/

            if (x == size - y - 1) {
                printf("\033[31m%5.2f\033[0m ", matrix[y][x]); // print in red
            } else {
                printf("%5.2f ", matrix[y][x]);
            }
        }
        printf("\n");
    }

    for (;;) { // no base case, just ctrl+c
        float needle;
        printf("\nSearch for an element in the antidiagonal: ");
        scanf("%f", &needle);

        int needle_y = size;
        // We're searching the antidiagonal. [y][size - y - 1]
        for (int y = size; y >= 0; y--) {
```

```

        if (roundf(needle * 100.0) == roundf(matrix[y][size - y - 1] * 100.0)) {
            needle_y = y;
            break;
        }
    }

    if (needle_y == size) {
        printf("No element could be found.");
    } else {
        printf("The first found element is (%d %d)", needle_y, size - needle_y -
1);
    }
}

return 0;
}

```

Результати тестування програми

Рандомні значення:

```

theamir in ~/projects/asd/lab/3 λ ./plain
Enter the matrix size (n*n): 10

14.38 72.55 31.96 38.93 78.50 57.05 68.57 17.92 91.93 91.99
95.21 39.92 80.48 16.91 72.27 72.92 21.49 44.13 33.12 5.82
52.86 89.83 56.17 15.11 60.22 95.44 14.15 81.94 63.88 59.32
19.46 78.26 32.87 51.42 18.19 12.37 9.48 86.75 30.29 2.41
79.74 26.50 42.33 61.22 43.41 15.60 35.14 64.90 59.74 68.26
70.72 13.60 59.09 27.89 28.70 20.30 24.33 42.85 3.25 88.21
3.17 22.71 67.46 36.04 74.13 85.65 48.41 83.61 73.41 78.70
86.02 54.15 6.19 29.35 16.36 49.60 44.95 51.50 15.50 5.69
20.76 86.22 19.29 79.84 15.10 47.99 1.15 39.43 90.85 4.40
28.64 94.02 27.11 96.10 31.06 2.24 82.75 79.46 85.85 57.16

Search for an element in the antidiagonal: 91.99
The first found element is (0 9)
Search for an element in the antidiagonal: 6.19
The first found element is (7 2)
Search for an element in the antidiagonal: 15.6
The first found element is (4 5)
Search for an element in the antidiagonal: 10
No element could be found.
Search for an element in the antidiagonal: 14.38
No element could be found.

```

```

theamir in ~/projects/asd/lab/3 λ ./plain
Enter the matrix size (n*n): 8

49.51 53.67 16.51 17.99 72.71 0.21 68.37 61.10
42.28 49.97 96.91 14.71 97.76 46.92 27.79 55.59
52.80 71.45 27.32 30.72 93.91 34.28 3.42 69.53
1.95 85.65 97.21 89.16 16.22 55.54 97.02 65.73
10.21 14.53 83.73 82.92 14.74 53.10 45.02 57.02
4.06 42.93 71.73 2.82 89.85 0.53 58.41 43.66
71.97 85.73 74.37 66.89 21.01 77.79 37.41 22.96
64.44 35.62 13.12 80.67 91.16 11.14 47.40 2.37

Search for an element in the antidiagonal: 61.1
The first found element is (0 7)
Search for an element in the antidiagonal: 64.44
The first found element is (7 0)
Search for an element in the antidiagonal: 16.2
No element could be found.
Search for an element in the antidiagonal: 16.22
The first found element is (3 4)
Search for an element in the antidiagonal: 0.21
No element could be found.

```

$A[y][x] = \text{abs}(x - y)$ для знаходження повторних значень:

```

theamir in ~/projects/asd/lab/3 λ ./plain
Enter the matrix size (n*n): 10

0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00
1.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00
2.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00
3.00 2.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00
4.00 3.00 2.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00
5.00 4.00 3.00 2.00 1.00 0.00 1.00 2.00 3.00 4.00
6.00 5.00 4.00 3.00 2.00 1.00 0.00 1.00 2.00 3.00
7.00 6.00 5.00 4.00 3.00 2.00 1.00 0.00 1.00 2.00
8.00 7.00 6.00 5.00 4.00 3.00 2.00 1.00 0.00 1.00
9.00 8.00 7.00 6.00 5.00 4.00 3.00 2.00 1.00 0.00

Search for an element in the antidiagonal: 7
The first found element is (8 1)
Search for an element in the antidiagonal: 1
The first found element is (5 4)
Search for an element in the antidiagonal: 5.5
No element could be found.
Search for an element in the antidiagonal: 10
No element could be found.
Search for an element in the antidiagonal: 

```

Висновки

Розв'язав задачу на лінійний пошук у двовимірному масиві, застосувавши алгоритм, що не ітерується по кожному елементу матриці.

Показав, що він працює на довільних дійсних числах, а також тоді, коли побічна діагональ містить числа, що повторюються.