

**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 IM-41
Димура Ілля Олександрович
Номер у списку групи: 7

Перевірила:

Молчанова А. А.

Київ 2024

Завдання

Оскільки екран монітора має площинні координати так само, як і двовимірний масив (матриця), але, на відміну від останнього, надає можливість візуально спостерігати виконання способу обходу, дана лабораторна робота виконується в координатах екрану монітора (дивись методичні вказівки до виконання даної лабораторної роботи).

2. Завданням даної лабораторної роботи є виконання заданого за варіантом способу обходу на екрані монітору в текстовому режимі, проставляючи довільний символ клавіатури (наприклад ‘*’) у порядку заданого способу обходу.

3. Оскільки при виводі символу у правий ніжній кут екрану відбувається зсув зображення на один рядок вгору (якщо тільки не використовується прямий доступ до відеопам'яті), останній рядок екрану монітора при виконанні завдання заповнювати не треба

Варіант 7: Задано матрицю дійсних чисел $A[m,n]$. У кожному рядку матриці знайти останній від'ємний елемент і його місце знаходження (координати)

Текст програми

```
#include <stdio.h>

int main(void) {
    int m, n;
    printf("number of rows: ");
    scanf("%d", &m);
    printf("number of columns: ");
    scanf("%d", &n);

    int matrix[m][n];
    for (int i = 0; i < m; i++) {
        printf("%d row of matrix (space separated): ", i + 1);
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
}
```

```
printf("input matrix:\n");

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

for (int i = 0; i < m; i++) {
    int min_elem_idx = -1;

    for (int j = n - 1; j ≥ 0; j--) {
        #ifdef DEBUG
            printf("[%d][%d] %d\n", i, j, matrix[i][j]);
        #endif

        if (matrix[i][j] < 0) {
            min_elem_idx = j;
            break;
        }
    }

    if (min_elem_idx == -1) {
        printf("row %d: negative element was not found\n", i);
    } else {
        printf("row %d: found negative element %d with index %d\n",
               i, matrix[i][min_elem_idx], min_elem_idx);
    }
}
```

```
    return 0;  
}
```

Оточення:

```
→ lab03 gcc --version  
Apple clang version 16.0.0 (clang-1600.0.26.3)  
Target: arm64-apple-darwin24.0.0  
Thread model: posix  
InstalledDir: /Applications/Xcode.app/Contents/D  
.xctoolchain/usr/bin  
→ lab03 gcc -o bin01 main.c  
→ lab03 ls bin*  
bin01
```

```
→ lab03 ./bin01
number of rows: 7
number of columns: 7
1 row of matrix (space separated): 88 1 67 34 -34 30 8
2 row of matrix (space separated): 125 133 113 101 49 42 -4
3 row of matrix (space separated): 33 31 68 40 132 66 -38
4 row of matrix (space separated): 20 -19 -38 28 149 76 42
5 row of matrix (space separated): -19 -9 104 50 87 68 132
6 row of matrix (space separated): 111 -32 125 80 77 43 -10
7 row of matrix (space separated): 144 -16 -25 -6 21 76 148
input matrix:
88 1 67 34 -34 30 8
125 133 113 101 49 42 -4
33 31 68 40 132 66 -38
20 -19 -38 28 149 76 42
-19 -9 104 50 87 68 132
111 -32 125 80 77 43 -10
144 -16 -25 -6 21 76 148
row 0: found negative element -34 with index 4
row 1: found negative element -4 with index 6
row 2: found negative element -38 with index 6
row 3: found negative element -38 with index 2
row 4: found negative element -9 with index 1
row 5: found negative element -10 with index 6
row 6: found negative element -6 with index 3
□
```

```
→ lab03 ./bin01
number of rows: 9
number of columns: 9
1 row of matrix (space separated): 8 76 133 -49 149 115 -47 96 -39
2 row of matrix (space separated): 26 14 84 -32 44 95 -39 49 105
3 row of matrix (space separated): -45 24 140 105 51 124 -35 45 -45
4 row of matrix (space separated): 41 -29 42 5 143 86 137 -16 55
5 row of matrix (space separated): -50 -22 108 10 -1 96 -23 107 141
6 row of matrix (space separated): 107 -40 102 -31 -45 27 -41 -9 70
7 row of matrix (space separated): 133 39 145 18 -14 38 7 130 70
8 row of matrix (space separated): 92 44 -44 129 65 13 15 66 75
9 row of matrix (space separated): 13 26 2 63 64 107 50 22 12
input matrix:
8 76 133 -49 149 115 -47 96 -39
26 14 84 -32 44 95 -39 49 105
-45 24 140 105 51 124 -35 45 -45
41 -29 42 5 143 86 137 -16 55
-50 -22 108 10 -1 96 -23 107 141
107 -40 102 -31 -45 27 -41 -9 70
133 39 145 18 -14 38 7 130 70
92 44 -44 129 65 13 15 66 75
13 26 2 63 64 107 50 22 12
row 0: found negative element -39 with index 8
row 1: found negative element -39 with index 6
row 2: found negative element -45 with index 8
row 3: found negative element -16 with index 7
row 4: found negative element -23 with index 6
row 5: found negative element -9 with index 7
row 6: found negative element -14 with index 4
row 7: found negative element -44 with index 2
row 8: negative element was not found
```

```
→ lab03 ./bin01
number of rows: 7
number of columns: 8
1 row of matrix (space separated): 28 106 142 98 -3 -30 111 83
2 row of matrix (space separated): 101 -41 139 6 38 45 26 3
3 row of matrix (space separated): 20 -26 -28 137 117 -35 1 76
4 row of matrix (space separated): -50 80 -32 148 100 29 140 93
5 row of matrix (space separated): 66 91 137 111 70 27 148 -41
6 row of matrix (space separated): 52 -30 124 27 132 98 6 -47
7 row of matrix (space separated): -26 96 57 -17 94 18 10 -38
input matrix:
28 106 142 98 -3 -30 111 83
101 -41 139 6 38 45 26 3
20 -26 -28 137 117 -35 1 76
-50 80 -32 148 100 29 140 93
66 91 137 111 70 27 148 -41
52 -30 124 27 132 98 6 -47
-26 96 57 -17 94 18 10 -38
row 0: found negative element -30 with index 5
row 1: found negative element -41 with index 1
row 2: found negative element -35 with index 5
row 3: found negative element -32 with index 2
row 4: found negative element -41 with index 7
row 5: found negative element -47 with index 7
row 6: found negative element -38 with index 7
```

```

row 9: found negative element -3 with index 9
→ lab03 ./script.sh 10 10
Input: 59 -4 44 17 143 59 150 92 81 28 85 -7 70 124 104 14 -1 19 -12 121 71 126 119 99 79 31 134 147 128 32 -44 77 -45 4
2 -4 88 2 84 -44 148 -10 31 25 64 143 -44 96 -32 57 34 -41 42 139 73 145 -24 149 104 -33 107 72 36 5 -1 6 20 9 65 50 -19
77 26 -34 80 26 -44 -43 37 80 109 67 48 -48 118 86 -32 60 135 142 28 -49 128 125 63 58 10 133 65 87 148

number of rows: number of columns: 1 row of matrix (space separated): 2 row of matrix (space separated): 3 row of matrix
(space separated): 4 row of matrix (space separated): 5 row of matrix (space separated): 6 row of matrix (space separat
ed): 7 row of matrix (space separated): 8 row of matrix (space separated): 9 row of matrix (space separated): 10 row of
matrix (space separated): input matrix:
59 -4 44 17 143 59 150 92 81 28
85 -7 70 124 104 14 -1 19 -12 121
71 126 119 99 79 31 134 147 128 32
-44 77 -45 42 -4 88 2 84 -44 148
-10 31 25 64 143 -44 96 -32 57 34
-41 42 139 73 145 -24 149 104 -33 107
72 36 5 -1 6 20 9 65 50 -19
77 26 -34 80 26 -44 -43 37 80 109
67 48 -48 118 86 -32 60 135 142 28
-49 128 125 63 58 10 133 65 87 148
row 0: found negative element -4 with index 1
row 1: found negative element -12 with index 8
row 2: negative element was not found
row 3: found negative element -44 with index 8
row 4: found negative element -32 with index 7
row 5: found negative element -33 with index 8
row 6: found negative element -19 with index 9
row 7: found negative element -43 with index 6
row 8: found negative element -32 with index 5
row 9: found negative element -49 with index 0

```

```

→ lab03 ./script.sh 10 10
Input: 38 92 44 112 -47 124 -45 150 130 137 -2 102 110 70 122 112 139 122 131 109 -8 -6 11 106 59 92 76 4 80 32 18 136 8
9 32 60 58 123 111 84 140 -50 143 112 60 135 150 4 -32 47 11 19 127 -13 133 22 145 18 -11 14 94 76 -8 -13 94 -40 -49 -35
128 136 110 19 -24 10 146 129 -14 -26 50 97 -36 59 34 -26 109 145 113 -30 113 147 -49 -23 88 108 130 13 79 25 1 131 15

number of rows: number of columns: 1 row of matrix (space separated): 2 row of matrix (space separated): 3 row of matrix
(space separated): 4 row of matrix (space separated): 5 row of matrix (space separated): 6 row of matrix (space separat
ed): 7 row of matrix (space separated): 8 row of matrix (space separated): 9 row of matrix (space separated): 10 row of
matrix (space separated): input matrix:
38 92 44 112 -47 124 -45 150 130 137
-2 102 110 70 122 112 139 122 131 109
-8 -6 11 106 59 92 76 4 80 32
18 136 89 32 60 58 123 111 84 140
-50 143 112 60 135 150 4 -32 47 11
19 127 -13 133 22 145 18 -11 14 94
76 -8 -13 94 -40 -49 -35 128 136 110
19 -24 10 146 129 -14 -26 50 97 -36
59 34 -26 109 145 113 -30 113 147 -49
-23 88 108 130 13 79 25 1 131 15
row 0: found negative element -45 with index 6
row 1: found negative element -2 with index 0
row 2: found negative element -6 with index 1
row 3: negative element was not found
row 4: found negative element -32 with index 7
row 5: found negative element -11 with index 7
row 6: found negative element -35 with index 6
row 7: found negative element -36 with index 9
row 8: found negative element -49 with index 9
row 9: found negative element -23 with index 0

```

Висновок:

Так як не було надано жодної інформації про взаємозв'язки елементів множини (наприклад, що значення відсортовані), прийшлося використати алгоритм лінійного пошуку. Вирішивши задачу пошуку заданої категорії елементів за допомогою алгоритму методу лінійного пошуку у двовимірних масивах, можна прийти до висновку, що в гіршому випадку складність алгоритму це $O(m * n)$. За умовою потрібно у кожному рядку матриці знайти останній від'ємний елемент, тому ефективніше було починати прохід з кінця рядка