

**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 №6**  
з дисципліни  
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-43  
Костеніч Степан Станіславович  
номер у списку групи: 17

Перевірила:

Молчанова А. А.

Київ 2024

## Постановка задачі

1. Задано двовимірний масив (матрицю) цілих чисел  $A[m,n]$  або  $A[n,n]$ , де  $m$  та  $n$  – натуральні числа (константи), що визначають розміри двовимірного масиву. Виконати сортування цього масиву або заданої за варіантом його частини у заданому порядку заданим алгоритмом (методом).

**Сортування повинно бути виконано безпосередньо у двовимірному масиві «на тому ж місці»**, тобто без перезаписування масиву та/або його будь-якої частини до інших одно- або двовимірних масивів, а також без використання спискових структур даних.

2. Розміри матриці  $m$  та  $n$  взяти самостійно у межах від 7 до 10.
3. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень матриці, щоб можна було легко відстежити коректність виконання сортування і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

## Варіант №17

Задано двовимірний масив (матрицю) цілих чисел  $A[m,n]$ . Відсортувати окремо кожен рядок масиву методом Шелла за незменшенням.

## Текст програми

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

#define M 9
#define N 10

int matrix[M][N] =
{
    {-83, -71, -43, -7, -7, -7, -7, 58, 69, 92}, //sorted
    {-94, -89, -81, -64, -62, -62, 15, 15, 15, 47}, //sorted
    { 97, 69, 59, 49, 49, 49, -43, -45, -70, -72}, //reverse-sorted
    { 89, 28, 26, 25, 8, 5, -20, -87, -87, -87}, //reverse-sorted
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 11, 11, 11, 11, 11, 11, 11, 11, 11, 11},
    { 17, 17, 17, 17, 17, -67, -67, -67, -67, -67}
};

int RandomRange(const int min, const int max) {
    return rand() % (max - min + 1) + min;
}

void FillRandom(const int startRow) {
    for (int i = startRow; i < M; i++) {
        for (int j = 0; j < N; j++) {
            matrix[i][j] = RandomRange(-100, 100);
        }
    }
}

void PrintMatrix() {
    printf("\n");
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            printf("%4d", matrix[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}

void MatrixShellSort() {
    for (int i = 0; i < M; i++) {
        int k;
```

```

        int T;

        for (int d = 4; d >= 1; d /= 2) {
            for (int j = 0; j < N; j++) {
                for (k = j, T = matrix[i][j]; k >= d && matrix[i][k - d]
> T; k -= d) {
                    matrix[i][k] = matrix[i][k - d];
                }
                matrix[i][k] = T;
            }
        }
    }

int main(void) {
    srand(time(nullptr));
    FillRandom(7);

    printf("Initial matrix");
    PrintMatrix();
    MatrixShellSort();
    printf("Sorted matrix");
    PrintMatrix();

    return 0;
}

```

## Результати тестування програми

```
Initial matrix
-83 -71 -43 -29 -7 20 38 58 69 92
-94 -89 -81 -64 -62 -62 -46 -35 15 47
 97  69  59  57  49  39 -43 -45 -70 -72
 89  28  26  25   8   5 -20 -21 -44 -87
 45  44 -16  86 -56  48  82 -99 -63  34
-75  15  83  15  62 -44  15 -81  79 -83
-24 -91 -22  45 -30  75  45 -33 -88 -59
-60 -50  44  84  75  84 -23  20 -19 -15
-61  22 -13 -53 -90 -63  22  71  40  23

Sorted matrix
-83 -71 -43 -29 -7 20 38 58 69 92
-94 -89 -81 -64 -62 -62 -46 -35 15 47
-72 -70 -45 -43 39 49 57 59 69 97
-87 -44 -21 -20  5  8 25 26 28 89
-99 -63 -56 -16 34 44 45 48 82 86
-83 -81 -75 -44 15 15 15 62 79 83
-91 -88 -59 -33 -30 -24 -22 45 45 75
-60 -50 -23 -19 -15 20 44 75 84 84
-90 -63 -61 -53 -13 22 22 23 40 71
```

Ромір матриці: **9 на 10**

Рядки 0, 1: **вже відсортовані.**

Рядки 2, 3: **обернено відсортовані.**

Рядки 4-8: **невідсортовані.**

Initial matrix									
-83	-71	-43	-7	-7	-7	-7	58	69	92
-94	-89	-81	-64	-62	-62	15	15	15	47
97	69	59	49	49	49	-43	-45	-70	-72
89	28	26	25	8	5	-20	-87	-87	-87
0	0	0	0	0	0	0	0	0	0
11	11	11	11	11	11	11	11	11	11
17	17	17	17	17	-67	-67	-67	-67	-67
60	37	-77	67	-17	-88	17	-72	-53	47
-50	-33	-38	-14	-18	19	13	21	71	34
Sorted matrix									
-83	-71	-43	-7	-7	-7	-7	58	69	92
-94	-89	-81	-64	-62	-62	15	15	15	47
-72	-70	-45	-43	49	49	49	59	69	97
-87	-87	-87	-20	5	8	25	26	28	89
0	0	0	0	0	0	0	0	0	0
11	11	11	11	11	11	11	11	11	11
-67	-67	-67	-67	-67	17	17	17	17	17
-88	-77	-72	-53	-17	17	37	47	60	67
-50	-38	-33	-18	-14	13	19	21	34	71

Ромір матриці: **9 на 10**

Рядки 0, 1: **вже відсортовані (з повтореннями).**

Рядки 2, 3: **обернено відсортовані (з повтореннями).**

Рядки 4, 5: **однакові числа**

Рядок 6: **обернено відсортовані, однакові.**

Рядки 7-8: **невідсортовані.**

Initial matrix									
70	-6	67	98	-46	-56	-25	-18	-62	-42
12	90	-41	-68	61	16	-43	96	-41	-58
91	36	78	-86	5	-32	4	-77	-16	-40
96	90	-40	-58	57	17	-88	52	-41	-5
97	-46	9	7	-84	-54	83	-35	98	-97
-61	-70	79	-55	-12	-32	6	-40	25	-93
-41	23	47	26	-1	-100	55	-46	86	-68
-34	-97	85	-53	-65	-64	92	-19	80	86
10	-86	-52	-29	-45	66	-16	4	-68	-56
Sorted matrix									
-62	-56	-46	-42	-25	-18	-6	67	70	98
-68	-58	-43	-41	-41	12	16	61	90	96
-86	-77	-40	-32	-16	4	5	36	78	91
-88	-58	-41	-40	-5	17	52	57	90	96
-97	-84	-54	-46	-35	7	9	83	97	98
-93	-70	-61	-55	-40	-32	-12	6	25	79
-100	-68	-46	-41	-1	23	26	47	55	86
-97	-65	-64	-53	-34	-19	80	85	86	92
-86	-68	-56	-52	-45	-29	-16	4	10	66

Ромір матриці: **9 на 10**

Рядки 0-8: **невідсортовані.**

Initial matrix

-30	-35	45	49	58	-42	34
94	90	11	-52	39	67	2
-67	-5	-47	36	42	-30	64
76	-92	24	-64	89	97	-32
10	-31	73	62	-58	-23	87
-94	74	-43	-90	33	35	17
-65	-3	72	17	-83	35	30

Sorted matrix

-42	-35	-30	34	45	49	58
-52	2	11	39	67	90	94
-67	-47	-30	-5	36	42	64
-92	-64	-32	24	76	89	97
-58	-31	-23	10	62	73	87
-94	-90	-43	17	33	35	74
-83	-65	-3	17	30	35	72

Ромір матриці: **7 на 7**

Рядки 0-8: **невідсортовані.**



Initial matrix						
40	-96	-30	64	-22	5	42
-27	-76	-61	-63	77	-27	-95
40	79	35	87	16	-95	98
7	-50	50	-31	62	39	38
-30	53	72	20	87	-86	-62
-58	27	75	23	34	-47	25
-100	-5	-67	18	25	-12	3
-19	-4	-33	-49	-30	95	74
-75	-92	-91	-95	-79	-90	89
Sorted matrix						
-96	-30	-22	5	40	42	64
-95	-76	-63	-61	-27	-27	77
-95	16	35	40	79	87	98
-50	-31	7	38	39	50	62
-86	-62	-30	20	53	72	87
-58	-47	23	25	27	34	75
-100	-67	-12	-5	3	18	25
-49	-33	-30	-19	-4	74	95
-95	-92	-91	-90	-79	-75	89

Ромір матриці: **9 на 7**

Рядки 0-8: **невідсортовані.**

## **Висновок**

Під час виконання лабораторної роботи № 6 я засвоїв теоретичний матеріал та набув практичних навичок рішення задач сортування заданої категорії елементів за допомогою різних алгоритмів сортування у двовимірних масивах.

Я поглибив свої знання щодо алгоритмів сортування, зокрема методу Шелла. Цей алгоритм є узагальненим варіантом алгоритму бульбашкового сортування, удосконаленого Шеллом за допомогою прямої вставки. Він відрізняється своєю ефективністю завдяки використанню принципу групування елементів, і ґрунтується на тому, що масив з частково відсортованими елементами сортується краще.

У ході виконання лабораторної роботи було реалізовано програму для сортування окремо кожного рядку масиву за незменшенням за допомогою алгоритму Шелла. Я більше дізнався про цей алгоритм сортування, на практиці глибше занурився, реалізував та використав його у масиві в поєднанні з обходом матриці по рядкам.

Отже, виконання лабораторної роботи № 6 було корисним, дозволило закріпити теоретичні знання та набути практичних навичок в області програмування мовою C.