

**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 №2.2**

з дисципліни  
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-32  
Король Олександр Володимирович  
номер у списку групи: 15

Перевірила:

Молчанова А. А.

Київ 2023

## 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 № 15

Задано квадратну двовимірний масив (матрицю) цілих чисел  $A[n,n]$ . Відсортувати побічну діагональ масиву алгоритмом №1 методу вставки (з лінійним пошуком зліва) за незменшенням.

## Текст програми

```
#include <stdio.h>

int main() {

    int n = 8;

    int i, j;

    int Array[8][8] = {

        {1 ,1 ,1, 1, 1, 1, 1, 4},

        {1, 1, 1, 1, 1, 1, 3, 1},

        {1, 1, 1, 1, 1, 6 ,1, 1},

        {1, 1, 1, 1, 8, 1, 1, 1},

        {1, 1, 1, 9, 1, 1, 1, 1},

        {1, 1, 1, 1, 1, 1, 1, 1},

        {1, 3, 1, 1, 1, 1, 1, 1},

        {2, 1, 1, 1, 1, 1, 1, 1},

    };

    for (i = 0; i < 8; i++) {

        for (j = 0; j < 8; j++) {

            printf("%d ", Array[i][j]);

        }

        printf("\n");

    }

    printf("\n");

    int current;

    for (i = 1; i < n; i++) {

        current = Array[(n-1)-i][i];

        j = 0;

        while (Array[(n-1)-j][j] < current) {
```

```
        j++;

    }

    for (int k = i-1; k >= j; k--) {

        Array[(n-1)-k-1][k+1] = Array[(n-1)-k][k];

    }

    Array[(n-1)-j][j] = current;

}

for (i = 0; i < 8; i++) {

    for (j = 0; j < 8; j++) {

        printf("%d ", Array[i][j]);

    }

    printf("\n");

}

return 0;

}
```

## Результати тестування програми

```
1 1 1 1 1 1 1 4
1 1 1 1 1 1 3 1
1 1 1 1 1 6 1 1
1 1 1 1 8 1 1 1
1 1 1 9 1 1 1 1
1 1 1 1 1 1 1 1
1 3 1 1 1 1 1 1
2 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 9
1 1 1 1 1 1 8 1
1 1 1 1 1 6 1 1
1 1 1 1 4 1 1 1
1 1 1 3 1 1 1 1
1 1 3 1 1 1 1 1
1 2 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

```
Process finished with exit code 0
```

```
1 1 1 1 1 1 1 9
1 1 1 1 1 1 8 1
1 1 1 1 1 7 1 1
1 1 1 1 6 1 1 1
1 1 1 5 1 1 1 1
1 1 4 1 1 1 1 1
1 3 1 1 1 1 1 1
2 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 9
1 1 1 1 1 1 8 1
1 1 1 1 1 7 1 1
1 1 1 1 6 1 1 1
1 1 1 5 1 1 1 1
1 1 4 1 1 1 1 1
1 3 1 1 1 1 1 1
2 1 1 1 1 1 1 1
```

```
Process finished with exit code 0
```

```
1 1 1 1 1 1 1 2
1 1 1 1 1 1 3 1
1 1 1 1 1 4 1 1
1 1 1 1 5 1 1 1
1 1 1 6 1 1 1 1
1 1 7 1 1 1 1 1
1 8 1 1 1 1 1 1
9 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 9
1 1 1 1 1 1 8 1
1 1 1 1 1 7 1 1
1 1 1 1 6 1 1 1
1 1 1 5 1 1 1 1
1 1 4 1 1 1 1 1
1 3 1 1 1 1 1 1
2 1 1 1 1 1 1 1
```

```
Process finished with exit code 0
```

```
81 61 51 61 51 12 11 2
91 21 21 11 14 13 3 13
61 35 25 13 14 4 21 23
91 41 11 13 5 51 21 15
91 23 21 6 14 51 21 13
61 43 7 1 11 31 31 19
32 8 81 14 15 11 11 18
9 11 17 31 19 51 15 17
```

```
81 61 51 61 51 12 11 9
91 21 21 11 14 13 8 13
61 35 25 13 14 7 21 23
91 41 11 13 6 51 21 15
91 23 21 5 14 51 21 13
61 43 4 1 11 31 31 19
32 3 81 14 15 11 11 18
2 11 17 31 19 51 15 17
```

```
Process finished with exit code 0
```



```
81 61 51 61 51 12 11 1252
91 21 21 11 14 13 5352 13
61 35 25 13 14 1231 21 23
91 41 11 13 2323 51 21 15
91 23 21 7567 14 51 21 13
61 43 2342 1 11 31 31 19
32 5232 81 14 15 11 11 18
1243 11 17 31 19 51 15 17
```

```
81 61 51 61 51 12 11 7567
91 21 21 11 14 13 5352 13
61 35 25 13 14 5232 21 23
91 41 11 13 2342 51 21 15
91 23 21 2323 14 51 21 13
61 43 1252 1 11 31 31 19
32 1243 81 14 15 11 11 18
1231 11 17 31 19 51 15 17
```

```
Process finished with exit code 0
```