

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 7 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Лінійні алгоритми»

Варіант 1

Виконав студент ІП-45 Янов Богдан Євгенійович

Перевірила старший викладач Вечерковська Анастасія Сергіївна

Київ 2024

## *Лабораторна робота №7*

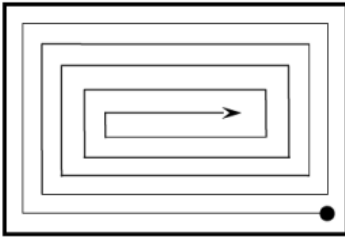
**Тема** - алгоритми сортування послідовностей..

**Мета** – дослідити алгоритми сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

### *Завдання*

Заповнити двовимірний масив.

Знайти значень елементів головної діагоналі двовимірного масиву та відсортувати їх обміном за зростанням.



### *Постановка задач*

1. Визначимо основні дії.
2. Деталізуємо дію ініціалізації початкових масивів
3. Деталізуємо дію заповнення двовимірного масиву (матриці)
4. Деталізуємо дію знаходження масиву діагональних елементів та їх сортування
5. Деталізуємо дію виводу масиву діагональних елементів

### *Математична модель*

Name	Type	Purpose
matrix	int[][]	Initial data
diag	int[]	Array of diagonal elements
N	#define	Size of column and row

N = 5

## *Псевдокод*

*Крок 1*

**Begin**

Initialize initial arrays

Fill matrix

Fill and sort diag

Print diag

**End**

*Крок 2*

**Begin**

matrix[N][N] = {0};

diag[N] = {0};

Fill matrix

Fill and sort diag

Print diag

**End**

*Крок 3*

**Begin**

matrix[N][N] = {0}

diag[N] = {0}

fillSpiral(matrix)

Fill and sort diag

Print diag

**End**

*Крок 4*

**Begin**

matrix[N][N] = {0}

diag[N] = {0}

fillSpiral(matrix)

fillDiagonal(matrix, diag)

sort(diag)

Print diag

**End**

*Крок 5*

**Begin**

```
matrix[N][N] = {0}
diag[N] = {0}
fillSpiral(matrix)
fillDiagonal(matrix, diag)
sort(diag)
printArray(diag)
```

**End**

*Допоміжні алгоритми*

**Begin** printArray(arr)

**Loop** for i from 0 to N

**Output** arr[i]

**End loop**

**End**

**Begin** fillSpiral(arr)

    top = 0

    bottom = N - 1

    left = 0

    right = N - 1

**While** top <= bottom && left <= right

**Loop** for i from right to left

            arr[bottom][i] = index++

**End loop**

        bottom--

**Loop** for i from bottom to top

            arr[i][left] = index++

**End loop**

        left++

**Loop** for i from left to right

            arr[top][i] = index++

**End loop**

        top++

**Loop** for i from top to bottom

            arr[i][right] = index++

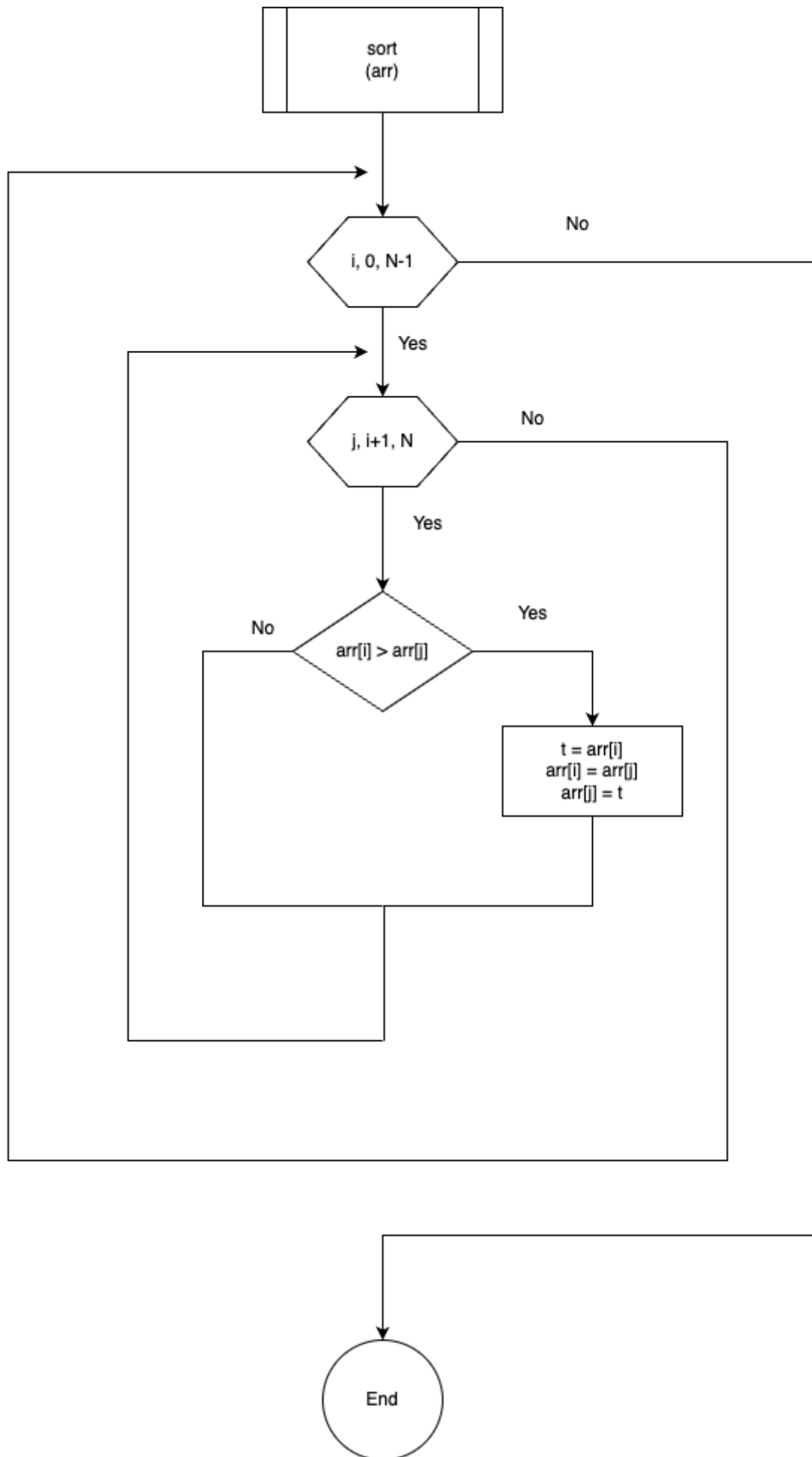
**End loop**

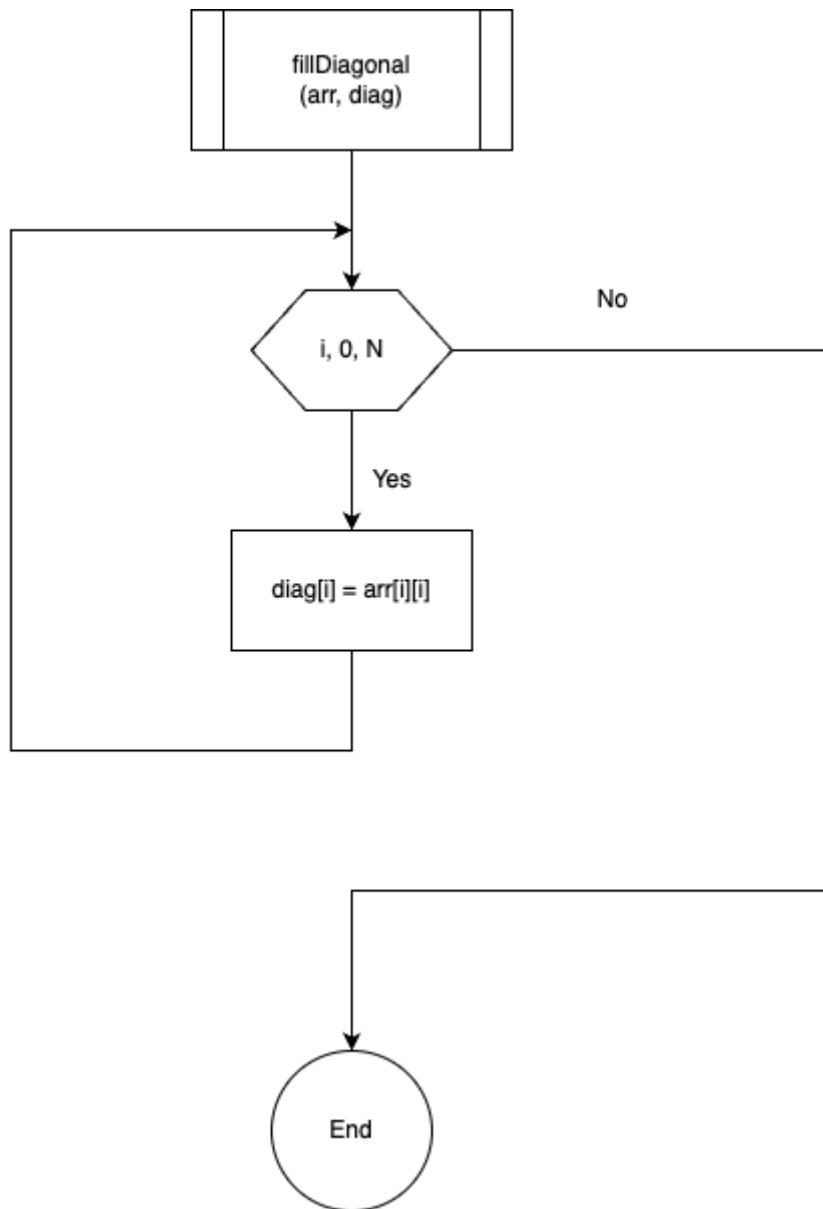
```
        right--
    End while
End

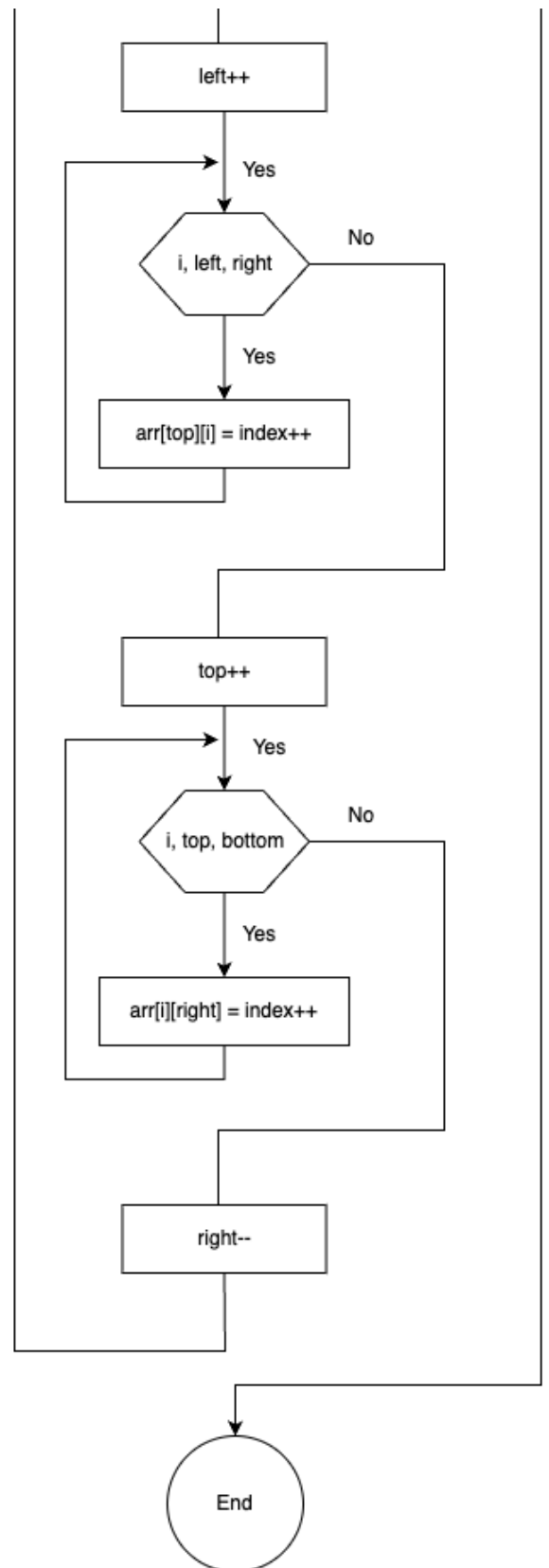
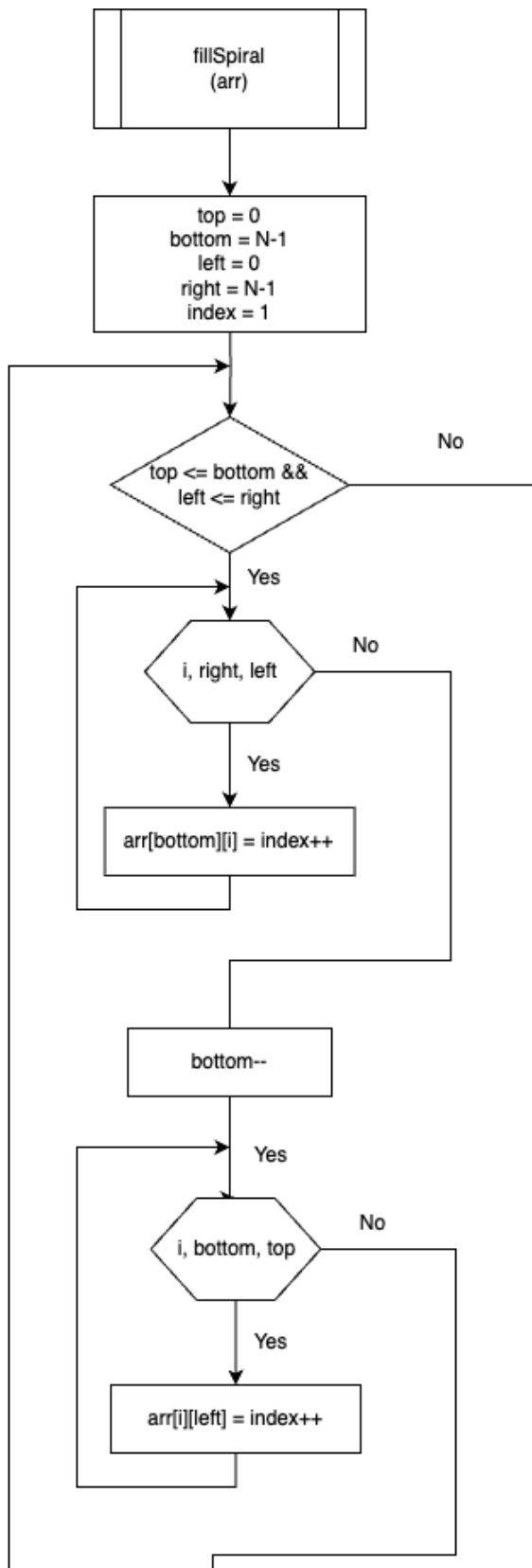
Begin sort(arr)
    Loop for i from 0 to N-1
        Loop for j from i+1 to N
            If arr[i] > arr[j]
                t = arr[i]
                arr[i] = arr[j]
                arr[j] = t
            End if
        End loop
    End loop
End

Begin fillDiagonal(arr, diag)
    Loop for i from 0 to N
        diag[i] = arr[i][i]
    End loop
End
```

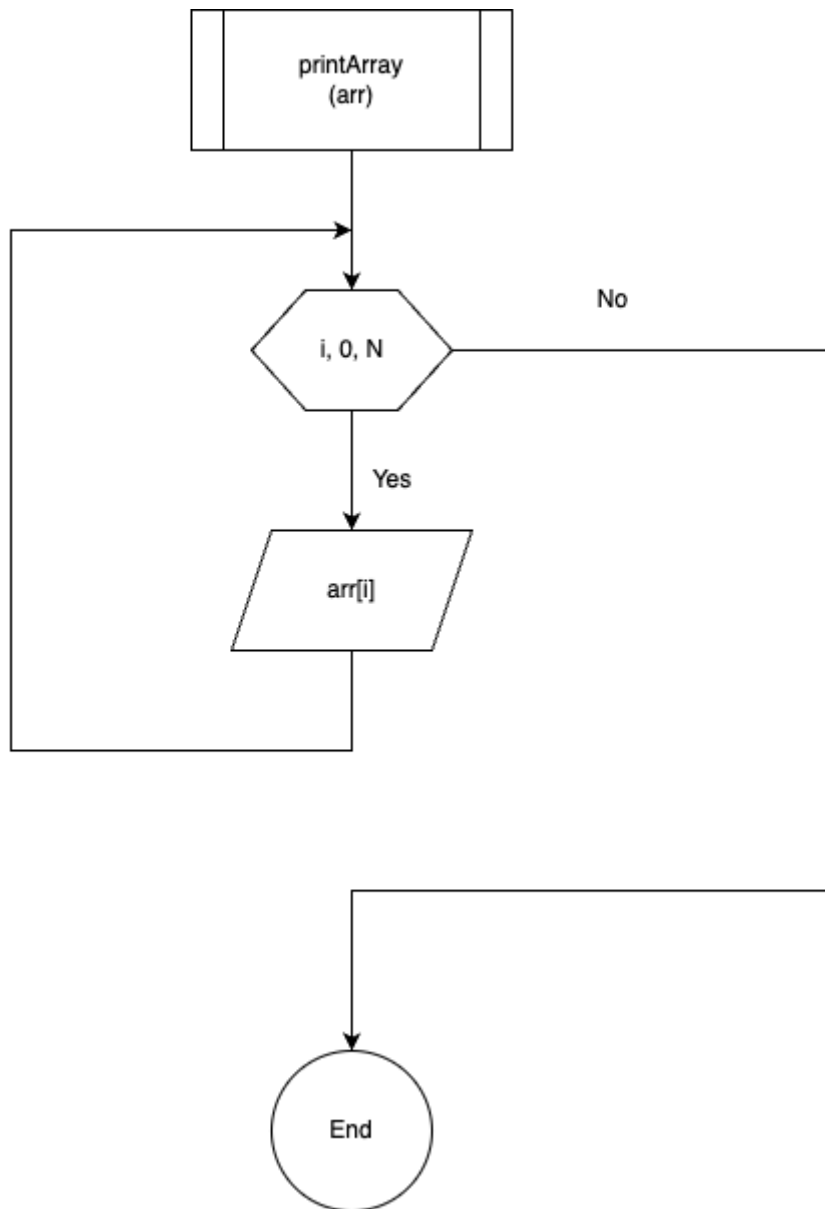
*Блок-схема*

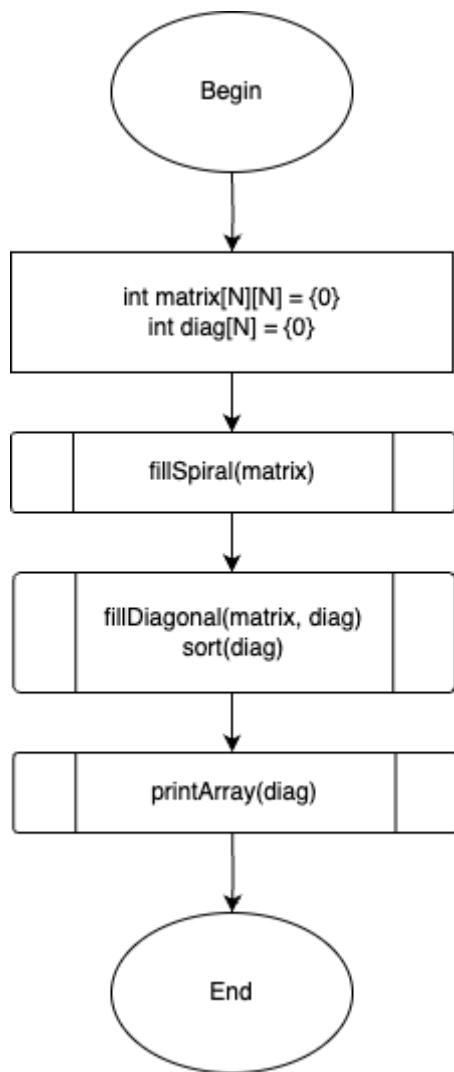












### *Тестування алгоритму*

Matrix:

<b>9</b>	10	11	12	13
8	<b>21</b>	22	23	14
7	20	<b>25</b>	24	15
6	19	18	<b>17</b>	16
5	4	3	2	<b>1</b>

Diagonal elements:

9	21	25	17	1
---	----	----	----	---

Sorted elements:

1	9	17	21	25
---	---	----	----	----

*Код мовою C*

```
#define N 5
```

```
void fillSpiral(int arr[N][N]) {  
    int top = 0, bottom = N - 1, left = 0, right = N - 1;  
    int index = 1;  
  
    while (top <= bottom && left <= right) {  
        for (int i = right; i >= left; i--)  
            arr[bottom][i] = index++;  
        bottom--;  
  
        for (int i = bottom; i >= top; i--)  
            arr[i][left] = index++;  
        left++;  
  
        for (int i = left; i <= right; i++)  
            arr[top][i] = index++;  
        top++;  
  
        for (int i = top; i <= bottom; i++)  
            arr[i][right] = index++;  
        right--;  
    }  
}
```

```

void fillDiagonal(int arr[N][N], int diag[N]) {
    for (int i = 0; i < N; i++) {
        diag[i] = arr[i][i];
    }
}

```

```

void sort(int arr[]) {
    for (int i = 0; i < N - 1; i++) {
        for (int j = i + 1; j < N; j++) {
            if (arr[i] > arr[j]) {
                int t = arr[i];
                arr[i] = arr[j];
                arr[j] = t;
            }
        }
    }
}

```

```

void printArray(int arr[]) {
    for (int i = 0; i < N; i++) {
        printf("%i ", arr[i]);
    }
    printf("\n");
}

```

```

int main() {
    int matrix[N][N] = {0};
    int diag[N] = {0};

    fillSpiral(matrix);
    fillDiagonal(matrix, diag);
    sort(diag);
}

```

```
    printArray(diag);  
  
    return 0;  
}
```

**Висновок:** Дослідив алгоритми сортування послідовностей та набув практичних навичок їх використання.