

**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-21
Сірик Максим Олександрович
номер у списку групи: 19

Перевірила:

Молчанова А. А.

Київ 2021

Завдання

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.

Вариант No 19

Задано квадратну двовимірний масив (матрицю) цілих чисел $A[n,n]$. Відсортувати головну діагональ масиву алгоритмом No2 методу вставки (з лінійним пошуком справа) за незменшенням.

Текст програми

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

```
void printMatrix(int n, int[][n]);
```

```
int main() {
```

```
printf("Lab 2.2\n");
```

```
// int a[][7] = {{7, 6, 4, 4, 3, 2, 1}, {7, 6, 4, 4, 3, 2, 1}, {7, 6,  
4, 4, 3, 2, 1}, {7, 6, 4, 4, 3, 2, 1}, {7, 6, 4, 4, 3, 2, 1}, {7, 6, 4,  
4, 3, 2, 1}, {7, 6, 4, 4, 3, 2, 1}, {};
```

```
// int a[][7] = {{1, 2, 3, 4, 4, 6, 7}, {1, 2, 3, 4, 4, 6, 7}, {1, 2, 3, 4, 4, 6, 7}, {1, 2, 3, 4, 4, 6, 7}, {1, 2, 3, 4, 4, 6, 7}, {1, 2, 3, 4, 4, 6, 7}, {1, 2, 3, 4, 4, 6, 7}};
```

```
int a[][7] = {{3, 8, 8, 5, 2, 6, 1}, {0, 0, 5, 2, 2, 5, 4}, {9, 4, 4, 5, 6, 8, 4}, {7, 8, 3, 8, 4, 9, 3}, {8, 5, 5, 7, 8, 3, 5}, {0, 2, 5, 0, 0, 6, 5}, {7, 7, 2, 7, 7, 5, 7}};
```

```
int n = sizeof(a) / sizeof(a[0]);
```

```
printMatrix(n, a);
```

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

```
    int j = i;
```

```
    int temp = a[j][j];
```

```
    while (temp < a[j-1][j-1] && j > 0) {
```

```
        a[j][j] = a[j-1][j-1];
```

```
        j--;
```

```
    }
```

```
    a[j][j] = temp;
```

```
}
```

```
printf("Sorted:\n");
```

```
printMatrix(n, a);
```

```
return 0;
```

```
}
```

```
void printMatrix(int n, int a[][n]) {
```

```
    for (int i = 0; i < n; i++) {
```

```
        for (int j = 0; j < n; j++)
```

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

```
        printf("\n");
```

```
}
```


2. Массив заповнений випадковими числами

```
→ gcc -o main main.c && ./main
Lab 2.2
3 8 8 5 2 6 1
0 0 5 2 2 5 4
9 4 4 5 6 8 4
7 8 3 8 4 9 3
8 5 5 7 8 3 5
0 2 5 0 0 6 5
7 7 2 7 7 5 7
Sorted:
0 8 8 5 2 6 1
0 3 5 2 2 5 4
9 4 4 5 6 8 4
7 8 3 6 4 9 3
8 5 5 7 7 3 5
0 2 5 0 0 8 5
7 7 2 7 7 5 8
```

3. Обернено відсортований масив

```
→ gcc -o main main.c && ./main
Lab 2.2
7 6 4 4 3 2 1
7 6 4 4 3 2 1
7 6 4 4 3 2 1
7 6 4 4 3 2 1
7 6 4 4 3 2 1
7 6 4 4 3 2 1
7 6 4 4 3 2 1
Sorted:
1 6 4 4 3 2 1
7 2 4 4 3 2 1
7 6 3 4 3 2 1
7 6 4 4 3 2 1
7 6 4 4 4 2 1
7 6 4 4 3 6 1
7 6 4 4 3 2 7
```