

**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  
Черепов Олександр Павлович  
номер у списку групи: 32

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

## Варіант № 2

Задано двовимірний масив (матрицю) цілих чисел  $A[m,n]$ . Відсортувати окремо кожен рядок масиву алгоритмом №2 методу вставки (з лінійним пошуком справа) за незбільшенням.

## Текст програми

```
#include <stdio.h>
int main(void) {
    int n, m, i, j, k;
    double T;

    printf("Enter the size of your matrix (2 values separated by space): <rows collumns>\n");
    scanf("%d %d", &m, &n);

    if ((m<7 || m>10) || (n<7 || n>10)) {
        printf("Error! The size values should be between 7 and 10.");
        return 0;
    }
}
```

```

double matrix[m][n];

for (i=0; i<m; i++) {
    for (j=0; j<n; j++) {
        printf("Enter a value for element [%d][%d]: ", i, j);
        scanf("%lf", &matrix[i][j]);
    }
}

printf("Input matrix:\n");
for (i=0; i<m; i++) {
    for (j=0; j<n; j++) {
        printf("%5.2lf\t", matrix[i][j]);
    }
    printf("\n");
}

for (i=0; i<m; i++){
    for (j=1; j<n; j++) {
        T = matrix[i][j];
        k = j;

        while (T>matrix[i][k-1] && k>0) {
            matrix[i][k] = matrix[i][k-1];
            k--;
        }

        matrix[i][k] = T;
    }
}

printf("Sorted matrix:\n");
for (i=0; i<m; i++) {
    for (j=0; j<n; j++) {
        printf("%5.2lf\t", matrix[i][j]);
    }
    printf("\n");
}

return 0;
}

```

# Тестування програми

## I тест (відсортований масив):

```
D:\education\EducationalWork\EducationalWork\ACD\000LABS\lab6\tests>test2.exe
Input matrix:
10.00  9.00  8.00  7.00  6.00  5.00  4.00  3.00
11.00  11.00  10.00  9.00  8.00  7.00  6.00  5.00
15.00  12.00  10.00  7.00  4.00  2.00  1.00  1.00
 5.00  4.00  3.00  2.00  1.00  0.00 -1.00 -2.00
10.00  5.00  2.50  1.25  0.00 -1.25 -2.50 -5.00
 5.00  5.00  5.00  5.00  5.00  5.00  1.00  1.00
22.00  20.00  18.00  16.00  14.00  12.00  10.00  8.00
Sorted matrix:
10.00  9.00  8.00  7.00  6.00  5.00  4.00  3.00
11.00  11.00  10.00  9.00  8.00  7.00  6.00  5.00
15.00  12.00  10.00  7.00  4.00  2.00  1.00  1.00
 5.00  4.00  3.00  2.00  1.00  0.00 -1.00 -2.00
10.00  5.00  2.50  1.25  0.00 -1.25 -2.50 -5.00
 5.00  5.00  5.00  5.00  5.00  5.00  1.00  1.00
22.00  20.00  18.00  16.00  14.00  12.00  10.00  8.00
```

## II тест (невідсортований масив):

```
D:\education\EducationalWork\EducationalWork\ACD\000LABS\lab6\tests>test1.exe
Input matrix:
 5.00  6.00  4.00  2.00  8.00  10.00  11.00
 1.00  7.00  3.00  9.00  2.00  5.00  4.00
 6.00  3.00  5.00  1.00  8.00  3.00  2.00
 8.00  5.00  3.00  1.00  4.00  6.00  7.00
10.00  3.00  1.00  7.00  8.00  4.00  2.00
11.00  12.00  4.00  7.00  2.00  3.00  1.00
 8.00  9.00  3.00  2.00  5.00  7.00  10.00
Sorted matrix:
11.00  10.00  8.00  6.00  5.00  4.00  2.00
 9.00  7.00  5.00  4.00  3.00  2.00  1.00
 8.00  6.00  5.00  3.00  3.00  2.00  1.00
 8.00  7.00  6.00  5.00  4.00  3.00  1.00
10.00  8.00  7.00  4.00  3.00  2.00  1.00
12.00  11.00  7.00  4.00  3.00  2.00  1.00
10.00  9.00  8.00  7.00  5.00  3.00  2.00
```

## III тест (обернено відсортований масив):

```
D:\education\EducationalWork\EducationalWork\ACD\000LABS\lab6\tests>test3.exe
Input matrix:
 1.00  2.00  3.00  4.00  5.00  6.00  7.00
 8.00  9.00  10.00  11.00  12.00  13.00  14.00
15.00  16.00  17.00  18.00  19.00  20.00  21.00
22.00  23.00  24.00  25.00  26.00  27.00  28.00
29.00  30.00  31.00  32.00  33.00  34.00  35.00
29.00  30.00  31.00  32.00  33.00  34.00  35.00
22.00  23.00  24.00  25.00  26.00  27.00  28.00
15.00  16.00  17.00  18.00  19.00  20.00  21.00
 8.00  9.00  10.00  11.00  12.00  13.00  14.00
 1.00  2.00  3.00  4.00  5.00  6.00  7.00
Sorted matrix:
 7.00  6.00  5.00  4.00  3.00  2.00  1.00
14.00  13.00  12.00  11.00  10.00  9.00  8.00
21.00  20.00  19.00  18.00  17.00  16.00  15.00
28.00  27.00  26.00  25.00  24.00  23.00  22.00
35.00  34.00  33.00  32.00  31.00  30.00  29.00
35.00  34.00  33.00  32.00  31.00  30.00  29.00
28.00  27.00  26.00  25.00  24.00  23.00  22.00
21.00  20.00  19.00  18.00  17.00  16.00  15.00
14.00  13.00  12.00  11.00  10.00  9.00  8.00
 7.00  6.00  5.00  4.00  3.00  2.00  1.00
```

## Висновок

На лабораторній роботі я розв'язував задачу по сортуванню рядків двовимірного масиву за незбільшенням, використовуючи Алгоритм№2, за допомогою мови програмування C. Для успішного виконання завдання я використовував Алгоритм№2, який обходить масив зліва направо та у випадку, якщо наступний елемент виявляється більшим за попередній, починається лінійний пошук комірки справа наліво для цього елемента та зсув всіх інших елементів вправо. На кожній ітерації циклу поточний елемент записувався в тимчасову комірку, тим самим звільняючи місце для інших. Відповідно, значення з цієї комірки порівнювалося з кожним елементом відсортованої частини масиву, рухаючись справа наліво, поки не було знайдено елемент, що є більшим за поточний. Одночасно з цим всі елементи до нього зсуваються, звільняючи місце, на яке стане елемент з тимчасової комірки.

У ході лабораторної роботи познайомився з труднощами сортування, а також з деякими алгоритмами сортування, зокрема Алгоритмом№2.

Отже, ця лабораторна робота була для мене надзвичайно важливою, адже її метою є навчитися сортуванню, яке є дуже корисним у вирішенні різних програмних задач.