

Факультет систем управління літальними апаратами  
Кафедра систем управління літальними апаратами

## XAI.301.312.7ЛР

Виконав студент гр. 312

07.05.2023  
(підпис, дата)

Зубов Євген Павлович  
(П.І.Б.)

## Перевірів

\_\_\_\_\_ к.т.н., доц. Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

## МЕТА РОБОТИ

Вивчити теоретичний матеріал по алгоритмам обробки масивів на мові C++, а також бібліотеки для роботи з файлами і реалізувати оголошення, введення з файлу, обробку і виведення в файл одновимірних і двовимірних масивів на мові C++ в середовищі Visual Studio.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. За допомогою текстового редактору створити текстовий файл «array\_in\_n.txt» з елементами вихідного масиву. У програмі на C++ перетворити масив відповідно до свого варіанту завдання, ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array\_out\_n.txt».

Array104. Дан масив розміру  $N$  і два цілих числа  $K$  і  $M$  ( $1 \leq K \leq N$ ,  $1 \leq M \leq 10$ ). Перед елементом масиву з номером  $K$  вставити  $M$  нових елементів з нульовими значеннями.

Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання розмір матриці і його елементи ввести з консолі. Вивести результати у консоль.

Matrix27. Дана матриця розміру  $M \times N$ . Знайти максимальний серед мінімальних елементів її рядків.

Завдання 3. Вивчити метод сортування відповідно до свого варіанту, проаналізувати його складність і продемонструвати на прикладі з 7-ми елементів (відповідно до свого варіанту). Реалізувати у вигляді окремої функції алгоритм сортування елементів масиву. Також окремими функціями реалізувати зчитування масиву з текстового файлу і виведення відсортованого масиву в консоль.

Sort8. Зробити сортування “Вибір” у порядку зменшення з типом даних “дійсний”

## ВИКОНАННЯ РОБОТИ

### Завдання 1.

Вирішення задачі Array104:

Вхідні дані (ім'я, опис, тип, обмеження):

array\_in\_104.txt - текстовий документ з розміром масиву та його даними

Вихідні дані (ім'я, опис, тип):

array\_out\_104.txt - текстовий документ, пустий

Екран роботи програми показаний на рис. Б.1-2

### Завдання 2.

Вирішення задачі Matrix27

Вхідні дані (ім'я, опис, тип, обмеження):

matrix\_in\_27.txt- текстовий документ з розміром масиву та його даними

M,N - стала, розмір масиву, int

Вихідні дані (ім'я, опис, тип):

matrix\_in\_27.txt - текстовий документ

Екран роботи програми показаний на рис. Б.3

Алгоритм вирішення показано на рис. 1

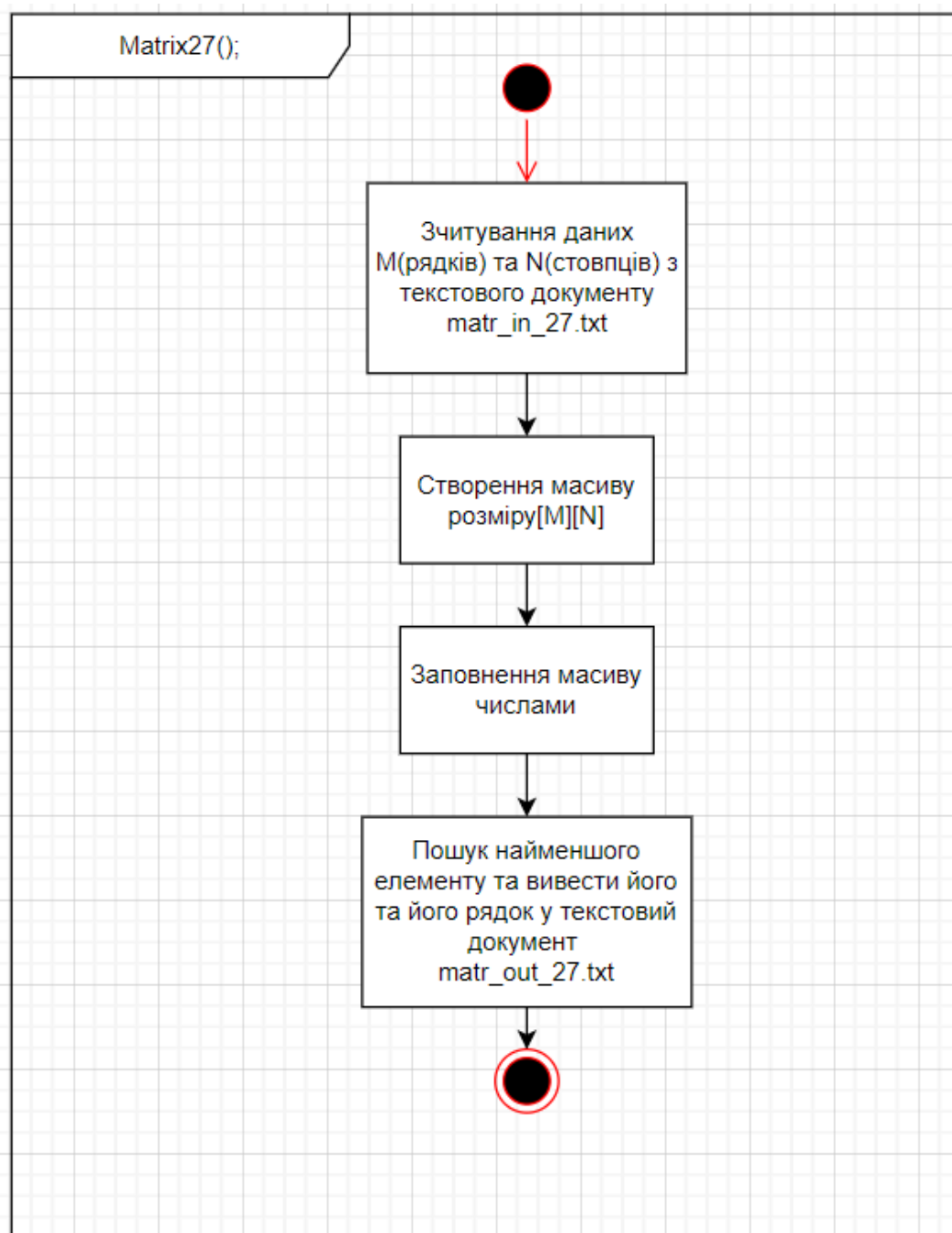


Рисунок 1 – Matrix27

Завдання 3.

Вирішення задачі Sort8

Вхідні дані (ім'я, опис, тип, обмеження):

sort\_in\_8.txt - текстовий файл з масивом

Вихідні дані (ім'я, опис, тип):

sort\_in\_8.txt - текстовий файл

Екран роботи програми показаний на рис. Б.4

Лістинг коду вирішення задач Array104, Matrix27 та Sort8 наведено в дод. А (стор. 6-14). Екран роботи програми показаний на рис. Б.1-4

## ВИСНОВКИ

Вивчено теоретичний матеріал по алгоритмам обробки масивів на мові С++, а також бібліотеки для роботи з файлами і реалізовано оголошення, введення з файлу, обробку і виведення в файл одновимірних і двовимірних масивів на мові С ++ в середовищі Visual Studio.

## ДОДАТОК А

## Лістинг коду програми до задач Matrix27 та Matrix63

```

#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
#include <ctime>

/*Array104. Дан масив розміру N і два цілих числа K і M ( $1 \leq K \leq N$ ,  $1 \leq M \leq 10$ ).
Перед елементом масиву з номером K вставити M нових елементів з нульовими
значеннями.*/

/*Matrix27. Дана матриця розміру M × N. Знайти максимальний серед
мінімальних
елементів її рядків.*/

/*Sort#8 Вибір Зменшення Дійсний*/

#define N 20
#define F 100

using namespace std;

void array104();
void fillArr(int*& arr, int lenght);
void get_nums(int size, int array[N]);
bool checkfiles(string in, string out);
void addNewInt(int*& arr, int n, string output, int K, int M);

void matrix27();
bool checkfile(string in);
void fillArray(string f_in, int**& arr, const int rows, const int cols);
void searchLowest(int** arr, const int rows, const int columns, string
f_out);

void sort8();
void choiceSort(float arr[N], int n, string in);

int main() {
    srand(time(NULL));

    int choice = 0;

    while (choice != 4) {

```

```

        cout << "Choose the task! : "
              "\n1.Array#104"
              "\n2.Matrix#27"
              "\n3.Sort#8"
              "\n4.Exit" << endl;

    cin >> choice;

    switch (choice) {
    case 1: {
        array104();          //Задание array104
        break;
    }
    case 2: {
        matrix27();          // Задание matrix27
        break;
    }
    case 3: {
        sort8();             // Задание sort8
        break;
    }
    case 4: {
        cout << "Program is end!";
        break;
    }
    default: {
        cout << "Wrong one, try again";
    }
    }
}

// Задание Array104

void array104() {

    int n = 0, startPos = 0, size = 0;
    int mas[N];

    get_nums(size, mas);
}

void get_nums(int size, int array[N]) {

    string filename_in = "array_in_104.txt";

```

```

string filename_out = "array_out_104.txt";

int startPos = 0;

ifstream f;

if (checkfiles(filename_in, filename_out)) {

    f.open(filename_in);

    string lenght;

    getline(f, lenght);
    int size = stoi(lenght);

    int* arr = new int(size);

    fillArr(arr, size);

    getline(f, lenght);

    int index = stoi(lenght);

    getline(f, lenght);

    int amountOfM = stoi(lenght);

    cout << "Index = " << index << "\n M = " << amountOfM;

    for (int i = 0; i < size; i++) {
        cout << "arr[" << i << "] = " << arr[i] << endl;
    }

    addNewInt(arr, size, filename_out, index, amountOfM);

}
else {
    cout << "Помилка, файл не знайдено";
}
}

void fillArr(int*& arr, int lenght) {
    for (int i = 0; i < lenght; i++) {
        arr[i] = rand() % 20;
    }
}

```



```

bool checkfiles(string in, string out) {

    ifstream f_in;
    ifstream f_out;

    f_in.open(in);
    f_out.open(out);

    if (!f_in.is_open() || !f_out.is_open()) {
        f_in.close();
        f_out.close();
        return 0;
    }
    else if (f_in.is_open() && f_out.is_open()) {
        return 1;
    }
    else {
        return 0;
    }
}

void addNewInt(int*& arr, int n, string filename_out, int K, int M) {

    cout << "loh";

    int save = n + M;

    int* temp = new int(save);

    for (int i = 0; i < K; i++) {
        temp[i] = arr[i];
    }

    for (int i = K; i < K + M + 1; i++) {
        temp[i] = 0;
    }

    for (int i = K + M; i < M + n; i++) {
        temp[i] = arr[i - M];
        K++;
    }

    cout << "New Array\n";

    for (int i = 0; i < save; i++) {
        cout << "Arr[" << i << "] = " << temp[i] << endl;
    }
}

```

```

ofstream outp(filename_out);
if (!outp)
{
    cerr << "Неможливо відкрити файл!" << endl;
    exit(1);
}
else {
    outp << "Новий масив: " << endl;
    for (int i = 0; i < save; i++) {
        outp << temp[i] << " ";
    }
}
}
// Конец Array104

// Задание Matrix27

void matrix27() {
    int n = 0, startPos = 0, rows = 0, cols = 0, skip = 0;
    string filename;

    string filename_in = "matrix_in_27.txt";

    ifstream f;

    if (checkfile(filename_in)) {
        f.open(filename_in);

        string size;

        getline(f, size);

        for (int i = 0; i < size.size(); i++) {
            if (size[i] == ' ') {
                skip = i;
            }
        }
        rows = stoi(size);
        if (rows > 20) {
            cout << "Рядків не може бути більше за 20" << endl;
            exit(0);
        }
        else {
            for (int i = 0; i < size.size() - skip; i++) {
                size[i] = size[skip + i];
            }
        }
    }
}

```

```

        size[skip + i] = 0;
    }

    cols = stoi(size);
    if (cols > 20) {
        cout << "Колон не може бути більше за 20" << endl;
        exit(0);
    }
    else {
        cout << "ROWS = " << rows << endl << "COLS = " <<
cols << endl;

        int** arr = new int* [rows];
        for (int i = 0; i < rows; i++) {
            arr[i] = new int[cols];
        }
        fillArray(filename_in, arr, rows, cols);
        searchLowest(arr, rows, cols, filename_in);
    }
}

else {
    cout << "Помилка, файл не знайдено";
}

}

void fillArray(string f_in, int**& arr, const int rows, const int cols) {
    string out;
    float num = 0;
    int numRows = 0, numCols = 0;

    ifstream f;

    istreamstringstream iss(out);
    f.open(f_in);

    std::getline(f, out);

    while (getline(f, out) && numRows < rows) {
        istreamstringstream iss(out);
        numCols = 0;

        while (iss >> num && numCols < cols) {
            arr[numRows][numCols] = num;
            numCols++;
        }
        numRows++;
    }
}

```

```

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                cout << arr[i][j] << " ";
            }
            cout << "\n";
        }
    }

    bool checkfile(string in) {

        ifstream f_in;
        f_in.open(in);

        if (f_in.is_open()) {
            return 1;
        }
        else {
            return 0;
        }
    }

    void searchLowest(int** arr, const int rows, const int columns, string
f_out) {
        int* lowest = new int[rows];
        int min = 0;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                if (arr[i][j] > min) {
                    lowest[i] = arr[i][j];
                }
            }
        }

        ofstream outp;
        outp.open(f_out, ios::app);

        if (!outp)
        {
            cerr << "Неможливо відкрити файл!" << std::endl;
            exit(1);
        }
        else {
            for (int i = 0; i < rows; i++) {
                outp << "\nRow #" << i << " equals " << lowest[i] << " ";
            }
        }
    }
}

```

```

    }
}

// Задание Sort8

void sort8() {
    const int M = 20;

    float arr[M];
    string filename;

    string filename_in = "sort_in_8.txt";

    ifstream f;
    istreamstream iss(filename_in);

    int count = 0;

    if (checkfile(filename_in)) {
        f.open(filename_in);

        while (count < M && f >> arr[count]) {
            count++;
        }

        for (int i = 0; i < count; i++) {
            cout << arr[i] << " ";
        }

        choiceSort(arr, count, filename_in);
    }
    else {
        cout << "Помилка, файл не знайдено";
    }
}

void choiceSort(float arr[N], int n, string f_in)
{
    for (int i = 0; i < n - 1; ++i) {
        int minIndex = i;
        for (int j = i + 1; j < n; ++j) {
            if (arr[j] > arr[minIndex]) {
                minIndex = j;
            }
        }
        if (minIndex != i) {

```

```

        std::swap(arr[i], arr[minIndex]);
    }
}

ofstream outp;
outp.open(f_in, ios::app);

if (!outp)
{
    cerr << "Неможливо відкрити файл!" << std::endl;
    exit(1);
}
else {
    outp << "\n";
    for (int i = 0; i < n; i++) {
        outp << arr[i] << " ";
    }
}
}

// Кінець Sort8

// Кінець

```

## ДОДАТОК Б

### Скріншоти вікна виконання програми

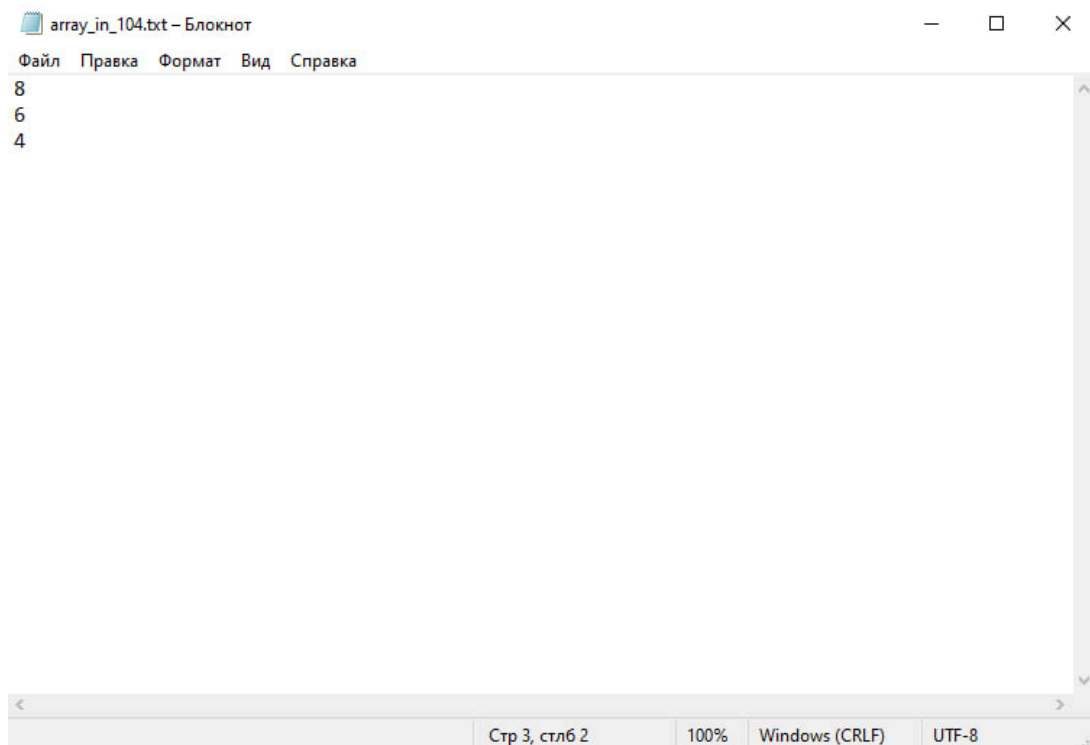


Рисунок Б.1

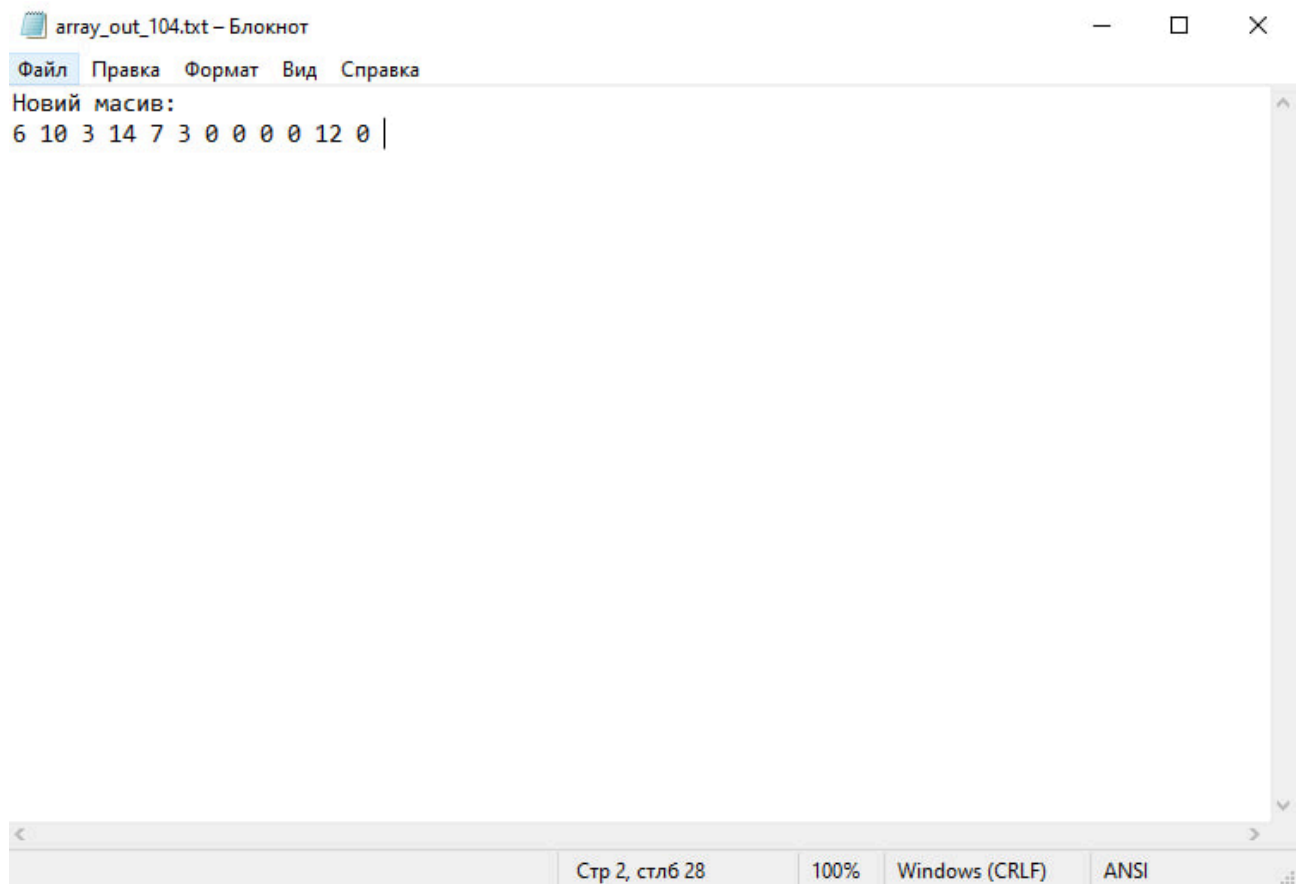


Рисунок Б.2

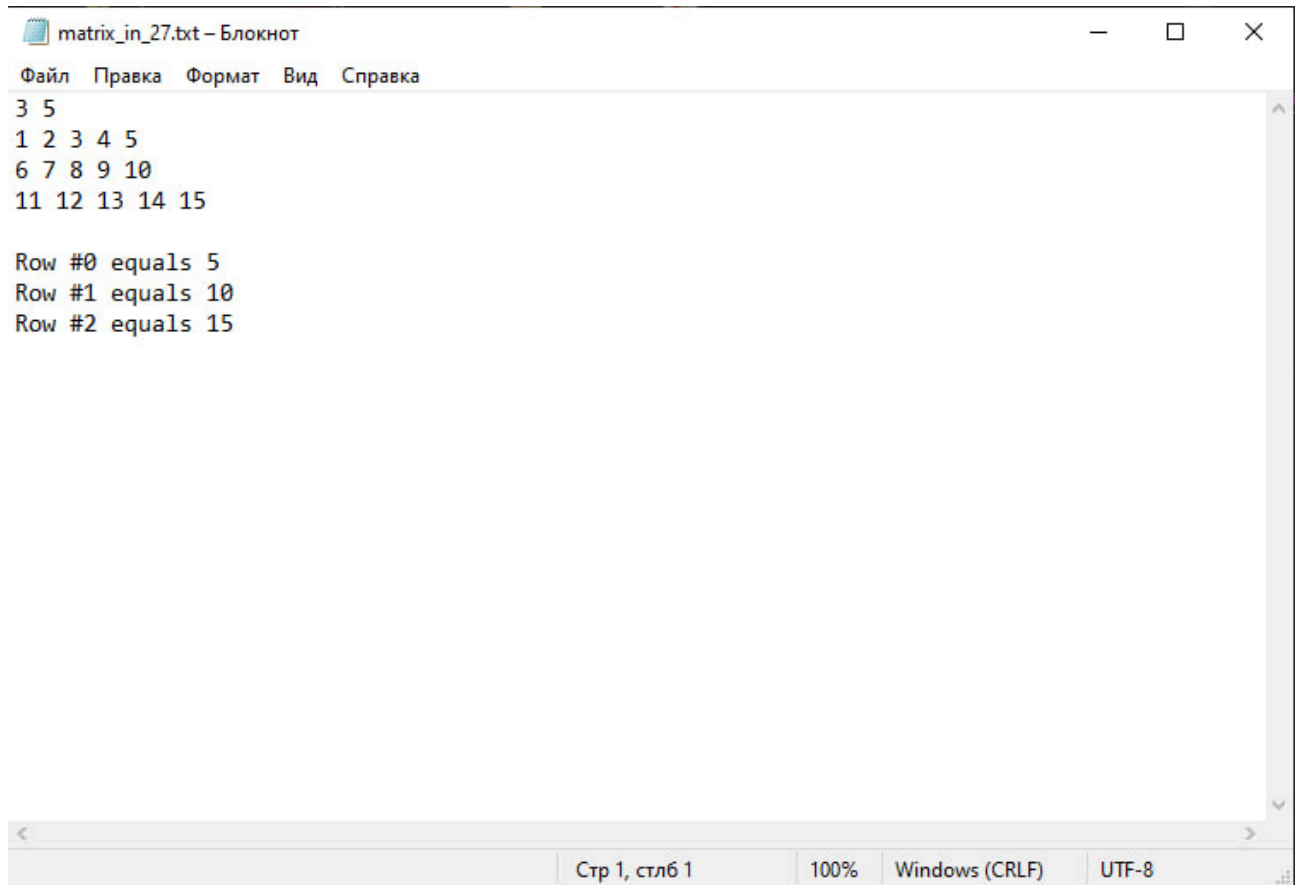


Рисунок Б.3

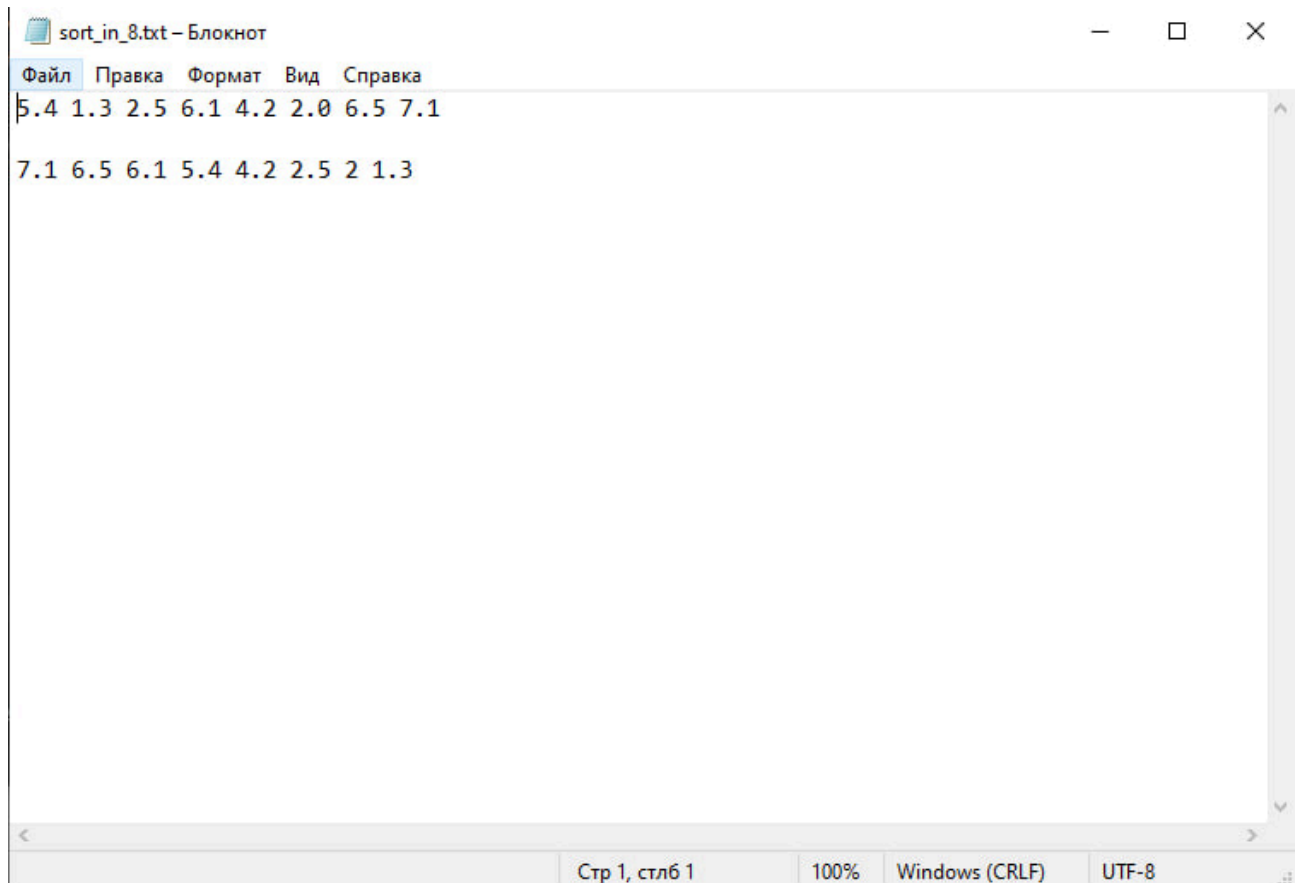




Рисунок Б.4