

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №8

за курсом “Програмування”

студентки групи ПА-23-1

Мазур Віолети

кафедра комп'ютерних технологій, ДНУ

2023/2024

1. Постановка задачі

Складіть програму, яка здійснює обробку вхідних даних за допомогою двовимірних масивів. При виконанні лабораторної роботи слід врахувати наступне:

1. Дати користувачеві можливість вибору: вводити вхідні дані з консолі, чи вводити їх з файлу; у випадку вводу з файлу продемонструвати дві можливості: а) можливість користувача задати ім'я файлу в діалозі; б) можливість перенапрявлення потоку вводу/виводу на файл в командному рядку.

2. При програмуванні однотипних підзадач (сортування, ввід/вивід масивів, знаходження суми стовпця матриці тощо) слід оформлювати їх у вигляді функцій.

Варіант 11. Задайте матриці A і B розмірністю $(n \times m)$, де n і $m \geq 6$. Виконайте такі завдання:

1. знайдіть кількість від'ємних елементів в обох матрицях;
2. визначте матрицю C як результат транспонування матриці A , якщо кількість від'ємних елементів матриці A більша кількості від'ємних елементів матриці B (в іншому випадку – як результат транспонування матриці B);
3. упорядкуйте за спаданням значення елементів заданого стовпця.

2. Опис розв'язку

Цей код виконує наступні операції:

1. Введення розміру матриць : Користувачу пропонується ввести кількість рядків та стовпців для матриць A і B. Цикли `do-while` перевіряють, чи введені значення більше або рівні 6.
 2. Вибір джерела даних : Користувачу пропонується вибрати, чи ввести дані з консолі або з файлу.
 3. Введення матриць : Якщо користувач обрав ввід з консолі, відбувається виклик функції `inputMatrixFromConsole`, яка заповнює матриці A та B даними, введеними з консолі. Якщо користувач обрав ввід з файлу, програма просить ввести ім'я файлу, після чого викликається функція `ifstream` для зчитування даних з файлу.
 4. Виведення матриць : Функція `printMatrix` виводить матриці A і B на екран.
 5. Знаходження кількості від'ємних елементів : Викликається функція `countNegativeElements` для обчислення кількості від'ємних елементів у матрицях A та B.
 6. Транспонування матриці : Викликається функція `transposeMatrix` для транспонування матриці A або B (залежно від того, у якій матриці більше від'ємних елементів) і запису результуючої матриці у transposed.
 7. Сортування стовпця за спаданням : Користувачеві пропонується ввести номер стовпця для сортування за спаданням. Здійснюється виклик функції `sortColumnDescending`, яка сортує вибраний стовпець матриці transposed.
- Цей код демонструє використання функцій для введення, виведення, обробки та маніпуляцій з матрицями.

3. Вихідний текст програми розв'язку задачі (основні фрагменти з коментарями)

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
// Функція для вводу матриці з консолі
```

```
void inputMatrixFromConsole(int** matrix, int rows, int cols) {  
    cout << "Enter the elements of the matrix:" << endl;  
    for (int i = 0; i < rows; ++i) {  
        matrix[i] = new int[cols]; // Виділення пам'яті для кожного рядка  
        for (int j = 0; j < cols; ++j) {  
            cout << "Enter element at position [" << i << "][" << j << "]: ";  
            cin >> matrix[i][j];  
        }  
    }  
}
```

// Функція для виводу матриці

```
void printMatrix(int** matrix, int rows, int cols) {  
    for (int i = 0; i < rows; ++i) {  
        for (int j = 0; j < cols; ++j) {  
            cout << setw(4) << matrix[i][j] << " ";  
        }  
        cout << endl;  
    }  
}
```

// Функція для знаходження кількості від'ємних елементів в матриці

```
int countNegativeElements(int** matrix, int rows, int cols) {  
    int count = 0;  
    for (int i = 0; i < rows; ++i) {  
        for (int j = 0; j < cols; ++j) {  
            if (matrix[i][j] < 0) {  
                count++;  
            }  
        }  
    }  
}
```

```

    }
}
}
return count;
}

```

// Функція для транспонування матриці

```

void transposeMatrix(int** matrix, int** transposed, int rows, int cols) {
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            transposed[j][i] = matrix[i][j];
        }
    }
}

```

// Функція для сортування стовпця за спаданням

```

void sortColumnDescending(int** matrix, int rows, int col) {
    for (int i = 0; i < rows - 1; ++i) {
        for (int j = 0; j < rows - i - 1; ++j) {
            if (matrix[j][col] < matrix[j + 1][col]) {
                swap(matrix[j][col], matrix[j + 1][col]);
            }
        }
    }
}

```

```

int main() {
    int rows, cols;

```

```

do {
    cout << "Enter the number of rows (should be 6 or more): ";
    cin >> rows;
    if (rows < 6) {
        cout << "Number of rows should be 6 or more. Please try again." << endl;
    }
} while (rows < 6);

do {
    cout << "Enter the number of columns (should be 6 or more): ";
    cin >> cols;
    if (cols < 6) {
        cout << "Number of columns should be 6 or more. Please try again." <<
endl;
    }
} while (cols < 6);

// Виділення пам'яті для матриць
int** matrixA = new int* [rows];
int** matrixB = new int* [rows];

// Вибір методу вводу даних
char choice;
cout << "Do you want to enter data from console? (y/n): ";
cin >> choice;

if (choice == 'y' || choice == 'Y') {
    inputMatrixFromConsole(matrixA, rows, cols);
}

```

```

        inputMatrixFromConsole(matrixB, rows, cols);
    }
else {
    // Введення даних з файлу
    string filename;
    cout << "Enter the name of the file: ";
    cin >> filename;

    ifstream file(filename);
    if (!file) {
        cerr << "Error: Unable to open file." << endl;
        return 1;
    }

    // Перевірка кількості елементів у файлі
    int expectedElements = (rows * cols) * 2;
    int actualElements = 0;

    // Введення даних для матриці A
    for (int i = 0; i < rows; ++i) {
        matrixA[i] = new int[cols]; // Виділення пам'яті для кожного рядка
        for (int j = 0; j < cols && !file.eof(); ++j) {
            file >> matrixA[i][j];
            actualElements++;
        }
    }

    // Введення даних для матриці B

```

```

    for (int i = 0; i < rows; ++i) {
        matrixB[i] = new int[cols]; // Виділення пам'яті для кожного рядка
        for (int j = 0; j < cols && !file.eof(); ++j) {
            file >> matrixB[i][j];
            actualElements++;
        }
    }
    file.close();
    if (actualElements < expectedElements) {
        cerr << "Error: Number of elements in the file is less than expected." <<
endl;
        return 1;
    }
}

// вивід матриць
cout << "Matrix A: " << endl;
printMatrix(matrixA, rows, cols);
cout << "Matrix B: " << endl;
printMatrix(matrixB, rows, cols);

// знаходження кількості від'ємних елементів в обох матрицях
int negativeCountA = countNegativeElements(matrixA, rows, cols);
int negativeCountB = countNegativeElements(matrixB, rows, cols);
cout << "Number of negative elements in matrix A: " << negativeCountA <<
endl;
cout << "Number of negative elements in matrix B: " << negativeCountB <<
endl;

```

```

// визначення матриці C як результат транспонування матриці A або B
int** transposed = new int* [cols];
for (int i = 0; i < cols; ++i) {
    transposed[i] = new int[rows];
}
if (negativeCountA > negativeCountB) {
    transposeMatrix(matrixA, transposed, rows, cols);
}
else {
    transposeMatrix(matrixB, transposed, rows, cols);
}
cout << "Transposed matrix C:" << endl;
printMatrix(transposed, cols, rows);

// сортування за спаданням значень елементів заданого стовпця
int columnToSort;

cout << "Enter the column number to sort in descending order (0 to " << cols
- 1 << "): ";
cin >> columnToSort;
if (columnToSort >= 0 && columnToSort < cols) {
    sortColumnDescending(transposed, rows, columnToSort);
    cout << "Sorted column " << columnToSort << " in descending order:" <<
endl;
    printMatrix(transposed, rows, cols);
}
else {
    cerr << "Error: Invalid column number." << endl;
}

```



```

// Звільнення пам'яті
for (int i = 0; i < rows; ++i) {
    delete[] matrixA[i];
    delete[] matrixB[i];
}
delete[] matrixA;
delete[] matrixB;

for (int i = 0; i < cols; ++i) {
    delete[] transposed[i];
}
delete[] transposed;

return 0;
}

```

4. Опис інтерфейсу програми (керівництво користувача)

Цей код розв'язує кілька завдань, пов'язаних з матрицями. Ось що він робить:

1. Користувач вводить кількість рядків і стовпців для матриць А та В. Якщо введені значення менші за 6, програма просить ввести їх знову.
2. Користувач обирає, чи хоче він ввести дані з консолі чи зчитати їх з файлу. Якщо він обирає файл, програма запитує назву файлу, з якого потрібно зчитати дані.
3. Якщо користувач обрав введення з консолі, він вводить елементи для кожної матриці. Якщо обрано зчитування з файлу, програма перевіряє, чи кількість елементів у файлі відповідає очікуваній кількості для обох матриць.
4. Після введення даних програма виводить обидві матриці на екран.

5. Потім програма рахує кількість від'ємних елементів у кожній матриці.
6. Після цього обирається матриця для транспонування: та, в якій більше від'ємних елементів.
7. Транспонується обрана матриця, і результат виводиться на екран.
8. Користувач вводить номер стовпця, який потрібно відсортувати за спаданням. Якщо введений номер неправильний, програма виводить повідомлення про помилку.
9. Обраний стовпець відсортовується за спаданням, і результат виводиться на екран.

5.Опис тестових прикладів

```
Введіть кількість рядків (min 6): 6
Введіть кількість стовпців (min 6): 6
Ви хочете ввести дані в консолі ? (y/n): y
Введіть елементи матриці:
Введіть елемент у позицію [0][0]: 45
Введіть елемент у позицію [0][1]: 6
Введіть елемент у позицію [0][2]: 8
Введіть елемент у позицію [0][3]: 2
Введіть елемент у позицію [0][4]: 4
Введіть елемент у позицію [0][5]: 5
Введіть елемент у позицію [1][0]: 6
Введіть елемент у позицію [1][1]: 7
Введіть елемент у позицію [1][2]: 8
Введіть елемент у позицію [1][3]: 9
Введіть елемент у позицію [1][4]: 8
Введіть елемент у позицію [1][5]: -7
Введіть елемент у позицію [2][0]: -4
Введіть елемент у позицію [2][1]: -2
Введіть елемент у позицію [2][2]: 678
Введіть елемент у позицію [2][3]: 345
Введіть елемент у позицію [2][4]: 23
Введіть елемент у позицію [2][5]: 6789
Введіть елемент у позицію [3][0]: 45
Введіть елемент у позицію [3][1]: 12
Введіть елемент у позицію [3][2]: 34
Введіть елемент у позицію [3][3]: 56
Введіть елемент у позицію [3][4]: 7
Введіть елемент у позицію [3][5]: 8
Введіть елемент у позицію [4][0]: 98
Введіть елемент у позицію [4][1]: 78
Введіть елемент у позицію [4][2]: 89
Введіть елемент у позицію [4][3]: 34
Введіть елемент у позицію [4][4]: 23
Введіть елемент у позицію [4][5]: 12
Введіть елемент у позицію [5][0]: -34
Введіть елемент у позицію [5][1]: -12
Введіть елемент у позицію [5][2]: -55
Введіть елемент у позицію [5][3]: -67
Введіть елемент у позицію [5][4]: 56
Введіть елемент у позицію [5][5]: 51
```

```
Введіть елементи матриці:
Введіть елемент у позицію [0][0]: 76
Введіть елемент у позицію [0][1]: 34
Введіть елемент у позицію [0][2]: 123
Введіть елемент у позицію [0][3]: 45
Введіть елемент у позицію [0][4]: 3
Введіть елемент у позицію [0][5]: 6
Введіть елемент у позицію [1][0]: 5
Введіть елемент у позицію [1][1]: 4
Введіть елемент у позицію [1][2]: 2
Введіть елемент у позицію [1][3]: 7
Введіть елемент у позицію [1][4]: 8
Введіть елемент у позицію [1][5]: 10
Введіть елемент у позицію [2][0]: 9
Введіть елемент у позицію [2][1]: 77
Введіть елемент у позицію [2][2]: 456
Введіть елемент у позицію [2][3]: -5
Введіть елемент у позицію [2][4]: -6
Введіть елемент у позицію [2][5]: -55
Введіть елемент у позицію [3][0]: 567
Введіть елемент у позицію [3][1]: -6
Введіть елемент у позицію [3][2]: -2
Введіть елемент у позицію [3][3]: 34
Введіть елемент у позицію [3][4]: 12
Введіть елемент у позицію [3][5]: 567
Введіть елемент у позицію [4][0]: 8999
Введіть елемент у позицію [4][1]: 4
Введіть елемент у позицію [4][2]: 23
Введіть елемент у позицію [4][3]: 6
Введіть елемент у позицію [4][4]: 7
Введіть елемент у позицію [4][5]: 8
Введіть елемент у позицію [5][0]: 9
Введіть елемент у позицію [5][1]: 3
Введіть елемент у позицію [5][2]: 4
Введіть елемент у позицію [5][3]: 2
Введіть елемент у позицію [5][4]: 3
Введіть елемент у позицію [5][5]: 4
```

```

Matrix A:
Матриця :
    45      6      8      2      4      5
    6       7      8      9      8     -7
   -4     -2     678    345    23    6789
    45     12     34     56     7      8
    98     78     89     34     23     12
   -34    -12    -55    -67     56     51

Matrix B:
Матриця :
    76     34    123     45     3      6
    5      4      2      7      8     10
    9     77    456    -5     -6    -55
   567    -6     -2     34     12    567
  8999     4     23      6      7      8
    9      3      4      2      3      4

Кількість негативних елементів у матриці A : 7
Кількість негативних елементів у матриці B: 5
Транспонована матриця C:
Матриця :
    45      6     -4     45     98    -34
    6       7     -2     12     78    -12
    8       8    678     34     89    -55
    2       9    345     56     34    -67
    4       8     23      7     23     56
    5     -7    6789      8     12     51

Введіть номер стовпця для сортування в порядку спадання (0 to 5): 3
Відсортований стовпець 3 у порядку спадання:
Матриця :
    45      6     -4     56     98    -34
    6       7     -2     45     78    -12
    8       8    678     34     89    -55
    2       9    345     12     34    -67
    4       8     23      8     23     56
    5     -7    6789      7     12     51

...Program finished with exit code 0
Press ENTER to exit console.

```

Рисунок 1.1 - 1.3. Робота програми

```

Введіть кількість рядків (min 6): 3
Введіть кількість стовпців (min 6): 3
Помилка : недостатня кількість рядків або стовпців.

...Program finished with exit code 1
Press ENTER to exit console.

```

Рисунок 1.4 - Робота програми, якщо дані введені не вірно ($n < 6$)

Введіть кількість рядків (min 6): 6	Введіть елемент у позицію [4][0]: -76
Введіть кількість стовпців (min 6): 9	Введіть елемент у позицію [4][1]: -43
Ви хочете ввести дані в консолі? (y/n): y	Введіть елемент у позицію [4][2]: -65
Введіть елементи матриці:	Введіть елемент у позицію [4][3]: -88
Введіть елемент у позицію [0][0]: 56	Введіть елемент у позицію [4][4]: -99
Введіть елемент у позицію [0][1]: 8	Введіть елемент у позицію [4][5]: -77
Введіть елемент у позицію [0][2]: 7	Введіть елемент у позицію [4][6]: 56
Введіть елемент у позицію [0][3]: 9	Введіть елемент у позицію [4][7]: 87
Введіть елемент у позицію [0][4]: 0	Введіть елемент у позицію [4][8]: 98
Введіть елемент у позицію [0][5]: -765	Введіть елемент у позицію [5][0]: 45
Введіть елемент у позицію [0][6]: 7777	Введіть елемент у позицію [5][1]: 56
Введіть елемент у позицію [0][7]: 33	Введіть елемент у позицію [5][2]: 76
Введіть елемент у позицію [0][8]: 65	Введіть елемент у позицію [5][3]: 88
Введіть елемент у позицію [1][0]: 4	Введіть елемент у позицію [5][4]: 3444
Введіть елемент у позицію [1][1]: 567	Введіть елемент у позицію [5][5]: 655
Введіть елемент у позицію [1][2]: 765	Введіть елемент у позицію [5][6]: 768
Введіть елемент у позицію [1][3]: 87	Введіть елемент у позицію [5][7]: 876
Введіть елемент у позицію [1][4]: 5	Введіть елемент у позицію [5][8]: 666
Введіть елемент у позицію [1][5]: 4	Введіть елементи матриці:
Введіть елемент у позицію [1][6]: 3	Введіть елемент у позицію [0][0]: 45
Введіть елемент у позицію [1][7]: 7	Введіть елемент у позицію [0][1]: 76
Введіть елемент у позицію [1][8]: 87	Введіть елемент у позицію [0][2]: 98
Введіть елемент у позицію [2][0]: 9999	Введіть елемент у позицію [0][3]: -6777
Введіть елемент у позицію [2][1]: 54	Введіть елемент у позицію [0][4]: 54
Введіть елемент у позицію [2][2]: 76	Введіть елемент у позицію [0][5]: 76
Введіть елемент у позицію [2][3]: 87	Введіть елемент у позицію [0][6]: 7
Введіть елемент у позицію [2][4]: 6543	Введіть елемент у позицію [0][7]: 8
Введіть елемент у позицію [2][5]: 45	Введіть елемент у позицію [0][8]: 45
Введіть елемент у позицію [2][6]: 65	Введіть елемент у позицію [1][0]: -6
Введіть елемент у позицію [2][7]: 85	Введіть елемент у позицію [1][1]: 5
Введіть елемент у позицію [2][8]: 75	Введіть елемент у позицію [1][2]: 4
Введіть елемент у позицію [3][0]: 333	Введіть елемент у позицію [1][3]: 3
Введіть елемент у позицію [3][1]: 65	Введіть елемент у позицію [1][4]: 2
Введіть елемент у позицію [3][2]: 76	Введіть елемент у позицію [1][5]: 1
Введіть елемент у позицію [3][3]: 98	Введіть елемент у позицію [1][6]: 11
Введіть елемент у позицію [3][4]: 99	Введіть елемент у позицію [1][7]: 222
Введіть елемент у позицію [3][5]: 90	Введіть елемент у позицію [1][8]: 33
Введіть елемент у позицію [3][6]: 95	Введіть елемент у позицію [2][0]: 44
Введіть елемент у позицію [3][7]: 4444	Введіть елемент у позицію [2][1]: 55
Введіть елемент у позицію [3][8]: -65	Введіть елемент у позицію [2][2]: 9
Введіть елемент у позицію [4][0]: -76	Введіть елемент у позицію [2][3]: 8
Введіть елемент у позицію [4][1]: -43	Введіть елемент у позицію [2][4]: 7
Введіть елемент у позицію [4][2]: -65	

```

Введіть елемент у позицію [2][6]: 5
Введіть елемент у позицію [2][7]: 99
Введіть елемент у позицію [2][8]: 55
Введіть елемент у позицію [3][0]: 66
Введіть елемент у позицію [3][1]: 77
Введіть елемент у позицію [3][2]: 88
Введіть елемент у позицію [3][3]: 87
Введіть елемент у позицію [3][4]: 67
Введіть елемент у позицію [3][5]: 56
Введіть елемент у позицію [3][6]: 98
Введіть елемент у позицію [3][7]: 79
Введіть елемент у позицію [3][8]: 65
Введіть елемент у позицію [4][0]: 4
Введіть елемент у позицію [4][1]: 6
Введіть елемент у позицію [4][2]: -6
Введіть елемент у позицію [4][3]: -7
Введіть елемент у позицію [4][4]: -88
Введіть елемент у позицію [4][5]: 0
Введіть елемент у позицію [4][6]: 23
Введіть елемент у позицію [4][7]: 44
Введіть елемент у позицію [4][8]: 67
Введіть елемент у позицію [5][0]: 34
Введіть елемент у позицію [5][1]: 56
Введіть елемент у позицію [5][2]: 7
Введіть елемент у позицію [5][3]: 8
Введіть елемент у позицію [5][4]: 7
Введіть елемент у позицію [5][5]: 6
Введіть елемент у позицію [5][6]: 5
Введіть елемент у позицію [5][7]: 4
Введіть елемент у позицію [5][8]: 43

```

Matrix A:

Матриця :

56	8	7	9	0	-765
4	567	765	87	5	4
9999	54	76	87	6543	45
333	65	76	98	99	90
-76	-43	-65	-88	-99	-77
45	56	76	88	3444	655
45	76	98	-6777	54	76
-6	5	4	3	2	1
44	55	9	8	7	6

Matrix B:

Матриця :

45	76	98	-6777	54	76
-6	5	4	3	2	1
44	55	9	8	7	6
66	77	88	87	67	56
4	6	-6	-7	-88	0
34	56	7	8	7	6
5	4	43	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Кількість негативних елементів у матриці A : 10

Кількість негативних елементів у матриці B : 7

Транспонована матриця C:

Матриця :

56	4	9999	333	-76	45
8	567	54	65	-43	56
7	765	76	76	-65	76
9	87	87	98	-88	88
0	5	6543	99	-99	3444
-765	4	45	90	-77	655
4	9999	333	-76	45	45
567	54	65	-43	56	76
765	76	76	-65	76	98

Введіть номер стовпця для сортування в порядку спадання (0 to 8): 7

Відсортований сталець 7 у порядку спадання:

Матриця :

56	4	9999	333	-76	45	8	9999	54
8	9999	54	65	-43	56	7	765	76
7	765	76	76	-65	76	9	567	87
9	567	87	98	-88	88	0	87	6543
0	87	6543	99	-99	3444	-765	5	45
-765	5	45	90	-77	655	4	4	333

...Program finished with exit code 0

Press ENTER to exit console.

Рисунок 1.5 - 1.8. Работа программы при значениях $n(\text{стов.}) - 9$, $n(\text{рядки}) - 6$

```
Enter the number of rows (should be 6 or more): 6
Enter the number of columns (should be 6 or more): 6
Do you want to enter data from console? (y/n): n
Enter the name of the file: input1.txt
Error: Number of elements in the file is less than expected.

C:\Users\Slim\Desktop\лаб 8\ConsoleApplication1\ConsoleApplication1
.exe (process 119996) exited with code 1.
Press any key to close this window . . .
```

```
Enter the number of rows (should be 6 or more): 6
Enter the number of columns (should be 6 or more): 6
Do you want to enter data from console? (y/n): n
Enter the name of the file: input4.txt
Matrix A:
  1  -2   3   4  -5   6
  7  -8   9 -10 -11  12
 13  14 -15  16  17  18
 19  20  21 -22 -23 -24
 25  26  27  28  29  30
 31  32  33  34  35  36
Matrix B:
 -1   2   3   4   5   6
  7   8   9 -10 -11 -12
-13 -14 -15 -16  17  18
 19  20  21  22  23  24
 25 -26 -27 -28 -29 -30
-31 -32 -33 -34 -35 -36
Number of negative elements in matrix A: 9
Number of negative elements in matrix B: 19
Transposed matrix C:
 -1   7 -13  19  25 -31
  2   8 -14  20 -26 -32
  3   9 -15  21 -27 -33
  4 -10 -16  22 -28 -34
  5 -11  17  23 -29 -35
  6 -12  18  24 -30 -36
Enter the column number to sort in descending order (0 to 5): 3
Sorted column 3 in descending order:
 -1   7 -13  24  25 -31
  2   8 -14  23 -26 -32
  3   9 -15  22 -27 -33
  4 -10 -16  21 -28 -34
  5 -11  17  20 -29 -35
  6 -12  18  19 -30 -36

C:\Users\Slim\Desktop\лаб 8\ConsoleApplication1\ConsoleApplication1\
.exe (process 117024) exited with code 0.
Press any key to close this window . . .
```

Рисунок 1.9 - 1.10. Работа программы з даними з файлу

6. Аналіз помилок (опис усунення зауважень)

Під час розробки програми виникло декілька помилок, які були усунені. Була помилка у логіці сортування, яка призводила до неправильного порядку елементів у вихідному масиві. Виправила умову порівняння у функції сортування для забезпечення правильного порядку елементів. Також спочатку були проблеми з читанням даних з файлу та записом результатів у файл. Перевірила правильність шляхів до файлів та виправила формат введення/виведення даних.

7. Висновки

У цій лабораторній роботі було поставлено завдання скласти програму, яка здійснює обробку вхідних даних за допомогою двовимірних масивів. При виконанні лабораторної роботи потрібно було врахувати наступне: можливість вибору (вводити вхідні дані з консолі, чи вводити їх з файлу; у випадку вводу з файлу продемонструвати дві можливості: а) можливість користувача задати ім'я файлу в діалозі; б) можливість перенапрявлення потоку вводу/виводу на файл в командному рядку.). При програмуванні однотипних підзадач (сортування, ввід/вивід масивів, знаходження суми стовпця матриці тощо) оформлювала їх у вигляді функцій. Потім задала матриці А і В розмірністю ($n \times m$), де n і $m \geq 6$ та виконала такі завдання: знайшла кількість від'ємних елементів в обох матрицях; визначила матрицю С як результат транспонування матриці А, якщо кількість від'ємних елементів матриці А більша кількості від'ємних елементів матриці В (в іншому випадку – як результат транспонування матриці В); упорядковувала за спаданням значення елементів заданого стовпця. Загалом, поставлені завдання допомогли більш точно навчитись аналізувати масиви за допомогою різних математичних операцій і методів сортування, покращуючи розуміння маніпулювання масивами та алгоритмічного вирішення проблем.