

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

У цьому кодi використовуються рiзні функції для роботи з матрицями. Ось що робить кожна частина коду:

1. Підключаються необхідні бібліотеки:

```
...  
  
#include <iostream>  
#include <fstream>  
#include <windows.h>  
...
```

2. Визначаються константи для розмірів матриць:

```
...  
  
const int MIN_ROWS = 6;  
const int MIN_COLS = 6;  
...
```

3. Оголошуються функції для введення матриці з консолі, виводу матриці, підрахунку від'ємних елементів, транспонування матриці та сортування стовпця за спаданням.

4. Оголошується функція `main`, яка виконує основний алгоритм програми.

5. У функції `main` встановлюються кодування вводу та виводу для підтримки кирилиці у консолі Windows:

```
...  
  
SetConsoleCP(1251);  
SetConsoleOutputCP(1251);  
...
```

6. Оголошуються дві матриці та змінні для розмірів матриць.

7. Користувачеві пропонується ввести кількість рядків та стовпців.

8. Користувачеві пропонується обрати метод введення даних: з консолі чи з файлу.

9. Якщо обраний варіант - введення з файлу, програма зчитує дані з файлу та заповнює матриці `matrixA` та `matrixB`.

10. Виводяться матриці `matrixA` та `matrixB`.

11. Підраховується кількість від'ємних елементів у кожній з матриць.
12. Визначається, яка матриця має більше від'ємних елементів, і транспонується відповідна матриця.
13. Транспонована матриця виводиться на екран.
14. Користувачеві пропонується ввести номер стовпця для сортування за спаданням.
15. Відповідний стовпець транспонованої матриці сортується за спаданням і виводиться на екран.
16. Програма завершується зі статусом 0.

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

```
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
const int MAX_ROWS = 6;
const int MAX_COLS = 6;

// Функція для вводу матриці з консолі
void inputMatrixFromConsole(int matrix[MAX_ROWS][MAX_COLS], int
rows, int cols) {
    cout << "Введіть елементи матриці:" << endl;
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            cout << "Введіть елемент у позицію [" << i << "][" << j << "]: ";
            cin >> matrix[i][j];
        }
    }
}
```

```
}
```

```
// Функція для виводу матриці
```

```
void printMatrix(int matrix[MAX_ROWS][MAX_COLS], int rows, int cols) {  
    cout << "Матриця :" << endl;  
    for (int i = 0; i < rows; ++i) {  
        for (int j = 0; j < cols; ++j) {  
            cout << setw(10) << matrix[i][j];  
        }  
        cout << endl;  
    }  
}
```

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

```
int countNegativeElements(int matrix[MAX_ROWS][MAX_COLS], 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[MAX_ROWS][MAX_COLS], int
transposed[MAX_ROWS][MAX_COLS], 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[MAX_ROWS][MAX_COLS], 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 matrixA[MAX_ROWS][MAX_COLS];

```

```

    int matrixB[MAX_ROWS][MAX_COLS];

```

```

    int rows, cols;

```

```

    cout << "Введіть кількість рядків (min 6): ";

```

```

cin >> rows;
cout << "Введіть кількість стовпців (min 6): ";
cin >> cols;

if (rows < MAX_ROWS || cols < MAX_COLS) {
    cerr << "Помилка : недостатня кількість рядків або стовпців." << endl;
    return 1;
}

// Вибір методу вводу даних
char choice;
cout << "Ви хочете ввести дані в консолі ? (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 << "Введіть назву файлу : ";
    cin >> filename;

    ifstream file(filename);
    if (!file) {
        cerr << "Помилка : неможливо відкрити файл." << endl;
        return 1;
    }
}

```

```
}
```

```
// Введення даних для матриці A
```

```
for (int i = 0; i < rows; ++i) {  
    for (int j = 0; j < cols; ++j) {  
        file >> matrixA[i][j];  
    }  
}
```

```
// Введення даних для матриці B
```

```
for (int i = 0; i < rows; ++i) {  
    for (int j = 0; j < cols; ++j) {  
        file >> matrixB[i][j];  
    }  
}  
file.close();  
}
```

```
// вивід матриць
```

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

```
// знаходження кількості від'ємних елементів в обох матрицях
```

```
int negativeCountA = countNegativeElements(matrixA, rows, cols);  
int negativeCountB = countNegativeElements(matrixB, rows, cols);  
cout << "Кількість негативних елементів у матриці A : " <<  
negativeCountA << endl;
```

```
cout << "Кількість негативних елементів у матриці B: " << negativeCountB  
<< endl;
```

```
// визначення матриці C як результат транспонування матриці A або B
```

```
int transposed[MAX_ROWS][MAX_COLS];
```

```
if (negativeCountA > negativeCountB) {
```

```
    transposeMatrix(matrixA, transposed, rows, cols);
```

```
}
```

```
else {
```

```
    transposeMatrix(matrixB, transposed, rows, cols);
```

```
}
```

```
cout << "Транспонована матриця C:" << endl;
```

```
printMatrix(transposed, cols, rows);
```

```
// сортування за спаданням значень елементів заданого стовпця
```

```
int columnToSort;
```

```
cout << "Введіть номер стовпця для сортування в порядку спадання (0 to  
" << cols - 1 << "): ";
```

```
cin >> columnToSort;
```

```
if (columnToSort >= 0 && columnToSort < cols) {
```

```
    sortColumnDescending(transposed, rows, columnToSort);
```

```
    cout << "Відсортований стовпець " << columnToSort << " у порядку  
спадання:" << endl;
```

```
    printMatrix(transposed, rows, cols);
```

```
}
```

```
else {
```

```
    cerr << "Помилка : недійсний номер стовпця." << endl;
```

```
}
```



```
    return 0;  
}
```

4. Опис інтерфейсу програми (керівництво користувача)

1. Введення кількості рядків та стовпців:

- Програма запитує користувача про кількість рядків та стовпців матриць.

2. Вибір методу вводу даних:

- Програма запитує користувача про метод введення даних: консоль (y/n).
- Якщо користувач обирає ввід з консолі, пропонується ввести елементи матриць вручну.

- Якщо користувач обирає ввід з файлу, програма запитує назву файлу та виконує зчитування даних з файлу.

3. Вивід матриць:

- Програма виводить обидві матриці на екран.

4. Підрахунок кількості від'ємних елементів:

- Програма підраховує кількість від'ємних елементів в обох матрицях та виводить результат.

5. Транспонування матриці:

- Програма визначає, яка з матриць має більше від'ємних елементів і транспонує її.

- Транспонована матриця виводиться на екран.

6. Сортування стовпця за спаданням:

- Користувач вводить номер стовпця, за яким буде виконане сортування.
- Програма сортує вибраний стовпець транспонованої матриці за спаданням та виводить результат.

7. Закінчення програми:

- Програма завершує свою роботу.

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 стовець 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 стовець 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. Робота програми при значеннях $n(\text{стов.}) - 9$, $n(\text{рядки}) - 6$

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 перенапрямлення 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 A і B розмірністю $(n \times m)$, де n і $m \geq 6$ та виконала такі завдання: знайшла кількість від'ємних елементів в обох матрицях; визначила 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); упорядкувала 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.