**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

Лабораторна робота №7  
*(вид роботи: лабораторна робота, індивідуальне завдання, курсова робота тощо)*

з дисципліни «Алгоритмізація та програмування»Тема роботи: «Двовимірні масиви даних»  
**Варіант № 1**

Виконав(-ла) студент(-ка)  
групи АнД - 11  
Яковкін Микола Андрійович

Перевірив(-ла):  
ПІП викладача

Київ – 2022

**Завдання 1**

**1.1 Математична постановка задачі (МПЗ).**

*Вхідні дані:*

K – значення цілого типу.

*Вихідні дані:*

Цілочиселньна матриця розміром ROWSxCOLS.

*Математична модель задачі:*

Введемо позначення. Константи:

ROWS – кількість рядків матриці.

COLS – кількість стовпців матриці.

Проміжні змінні:

i, j – цілі, змінна для індексів рядка та стовпця.

Для і = 0, ROWS-1:

Для j = 0, COLS-1 :

A[i][j] = K \* i;

Подання математичної постановки задачі у вигляді таблиці:

Таблиця 1.1

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| K – значення цілого типу. | Для і = 0, ROWS-1:  Для j = 0, COLS-1 :  A[i][j] = K \* i;  Виведення A[i][j]; | Цілочиселньна матриця розміром ROWSxCOLS. |
| **Константи:**  ROWS – кількість рядків матриці.  COLS – кількість стовпців матриці.  **Проміжні змінні:**  i, j – цілі, змінна для індексів рядка та  стовпця. |

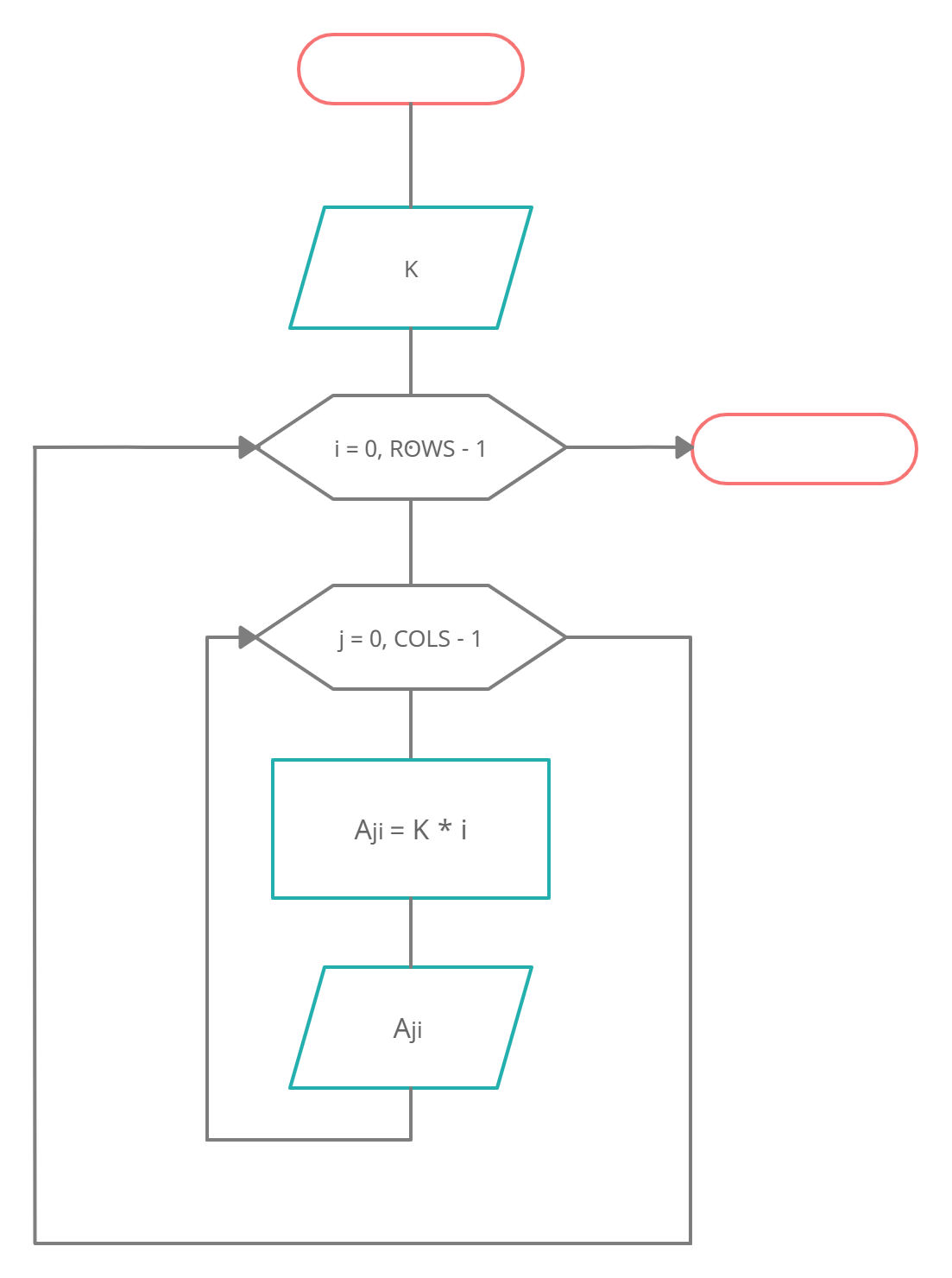
****

Рисунок 1.1 – Схема алгоритму для задачі 1

**1.2 Тестування програми**

Для доведення працездатності роботи програми проведемо таку перевірку:

ROWS = 4, COLS = 5 та К = 5. Маємо отримати:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 5 | 5 | 5 | 5 | 5 |
| 10 | 10 | 10 | 10 | 10 |
| 15 | 15 | 15 | 15 | 15 |

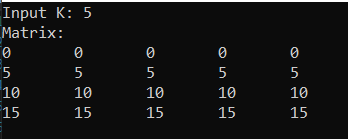


Рисунок 1.2 – Тестування програми.

**1.3 Текст програмною мовою С++.**

#**include** <iostream>

#**define** ROWS 4

#**define** COLS 5

**using namespace** std;

**void** main()

{

**int** matrix[ROWS][COLS];

**int** K;

cout << "Input K: ";

cin >> K;

cout << "Matrix: " << endl;

**for** (**int** i = 0; i < ROWS; i++)

{

**for** (**int** j = 0; j < COLS; j++)

{

matrix[i][j] = K \* i;

cout << matrix[i][j] << '\t';

}

cout << endl;

}

}

**Завдання 2**

**2.1 Математична постановка задачі (МПЗ).**

*Вхідні дані:*

c, p – границі цілого типу.

*Вихідні дані:*

sum – сума парних елементів матриці, цілого типу.

*Математична модель задачі:*

Введемо позначення. Константи:

ROWS – кількість рядків матриці.

COLS – кількість стовпців матриці.

Проміжні змінні:

i, j – цілі, змінна для індексів рядка та стовпця.

Для і = 0, ROWS-1:

Для j = 0, COLS-1 :

Якщо {A[i][j] / 2} = 0:

Якщо A[i][j] > p або A[i][j] < c:

sum = sum + A[i][j]

Подання математичної постановки задачі у вигляді таблиці:

Таблиця 2.1

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| c, p – границі цілого типу. | Для і = 0, ROWS-1:  Для j = 0, COLS-1 :  Якщо {A[i][j] / 2} = 0:  Якщо A[i][j] > p або A[i][j] < c:  sum = sum + A[i][j] | sum – сума парних елементів матриці, цілого типу. |
| **Константи:**  ROWS – кількість рядків матриці.  COLS – кількість стовпців матриці.  **Проміжні змінні:**  i, j – цілі, змінна для індексів рядка та  стовпця. |

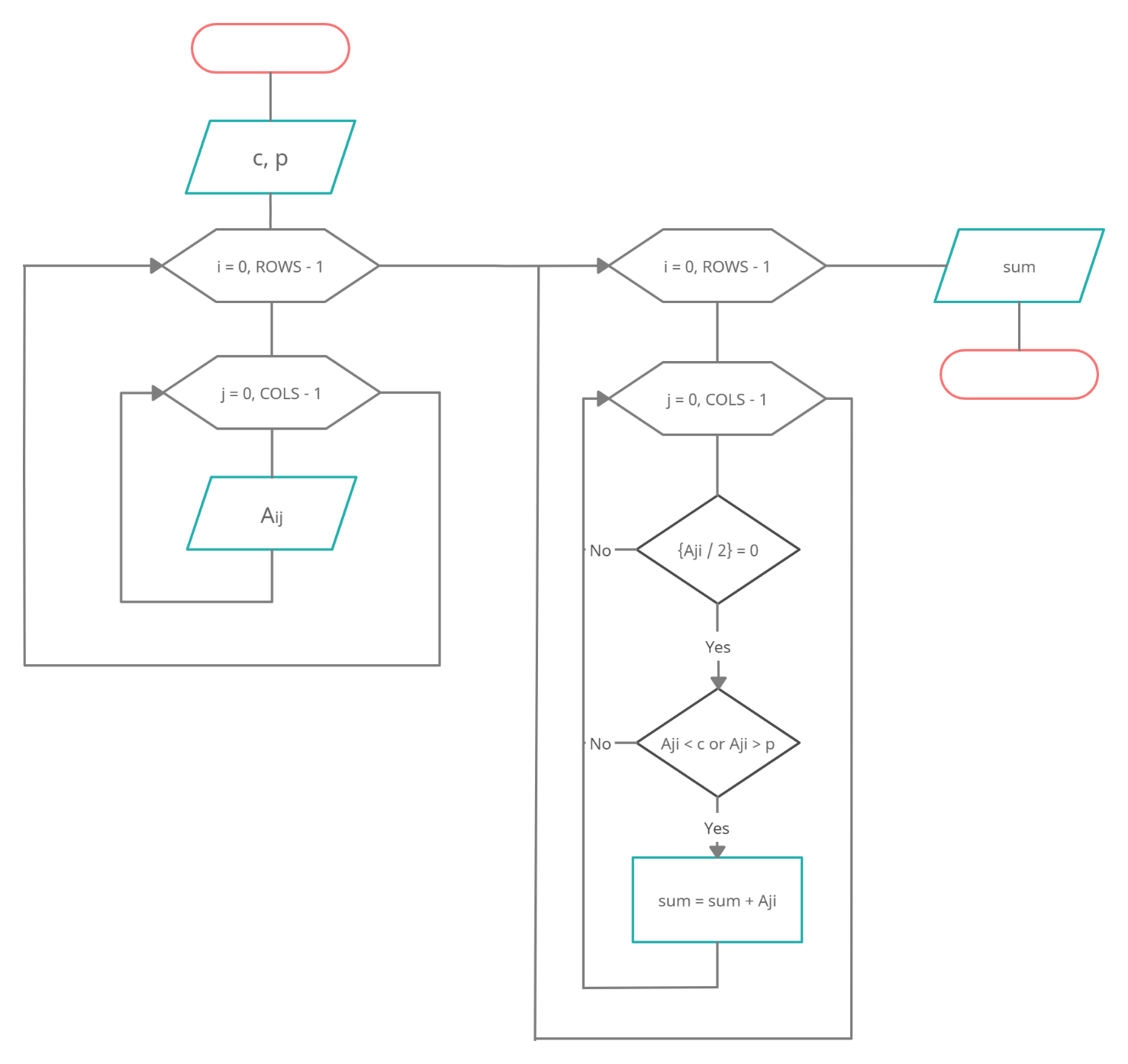


Рисунок 2.1 - Схема алгоритму для задачі 2.

**2.2 Тестування програми**

Для доведення працездатності роботи програми проведемо таку перевірку:

ROWS = 4, COLS = 5, c = 3, p = 6 та матрицю:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 7 | 4 | 0 | 9 |
| 4 | 8 | 8 | 2 | 4 |
| 5 | 5 | 1 | 7 | 1 |
| 1 | 5 | 2 | 7 | 6 |

Маємо отримати: sum = 20.

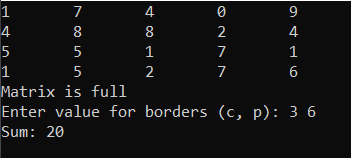


Рисунок 2.2 – Тестування програми.

**2.3 Текст програмною мовою С++.**

#**include** <iostream>

#**define** ROWS 4

#**define** COLS 5

**using namespace** std;

//**void** Initialize(int[ROWS][COLS], int, int);

**void** main()

{

**int** matrix[ROWS][COLS];

**int** borderC, borderP;

**int** sum = 0;

//Initialize(matrix, ROWS, COLS);

**for** (**int** i = 0; i < ROWS; i++)

{

**for** (**int** j = 0; j < COLS; j++)

{

matrix[i][j] = rand() % 10;

cout << matrix[i][j] << '\t';

}

cout << endl;

}

cout << "Matrix is full" << endl;

cout << "Enter value for borders (c, p): ";

cin >> borderC >> borderP;

**for** (**int** i = 0; i < ROWS; i++)

{

**for** (**int** j = 0; j < COLS; j++)

{

**if** (matrix[i][j] % 2 == 0)

{

**if** (matrix[i][j] < borderC || borderP < matrix[i][j])

{

//cout << matrix[i][j] << endl;

sum += matrix[i][j];

}

}

}

}

cout << "Sum: " << sum << endl;

}

**Завдання 3**

**3.1 Математична постановка задачі (МПЗ).**

*Вхідні дані:* -

*Вихідні дані:*

Максимум для кожного рядка цілого типу.

Мінімум для кожного рядка цілого типу.

*Математична модель задачі:*

Введемо позначення. Константи:

ROWS – кількість рядків матриці.

COLS – кількість стовпців матриці.

Проміжні змінні:

i, j – цілі, змінна для індексів рядка та стовпця.

minValue – мінімальне значення (100).

maxValue – максимальне значення (0).

Для і = 0, ROWS-1:

maxValue = 0

Для j = 0, COLS-1 :

Якщо Aij > maxValue: maxValue = Aij

Для j = 0, ROWS-1:

minValue = 100

Для i = 0, COLS-1 :

Якщо Aji > minValue: minValue = Aij

Подання математичної постановки задачі у вигляді таблиці:

Таблиця 3.1

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| Нічого | Для і = 0, ROWS-1:  maxValue = 0  Для j = 0, COLS-1 :  Якщо Aij > maxValue: maxValue = Aij  Для j = 0, ROWS-1:  minValue = 100  Для i = 0, COLS-1 :  Якщо Aji > minValue: minValue = Aij | Максимум для кожного рядка цілого типу.  Мінімум для кожного рядка цілого типу. |
| **Константи:**  ROWS – кількість рядків матриці.  COLS – кількість стовпців матриці.  **Проміжні змінні:**  i, j – цілі, змінна для індексів рядка та  стовпця.  minValue – мінімальне значення (100).  maxValue – максимальне значення (0). |

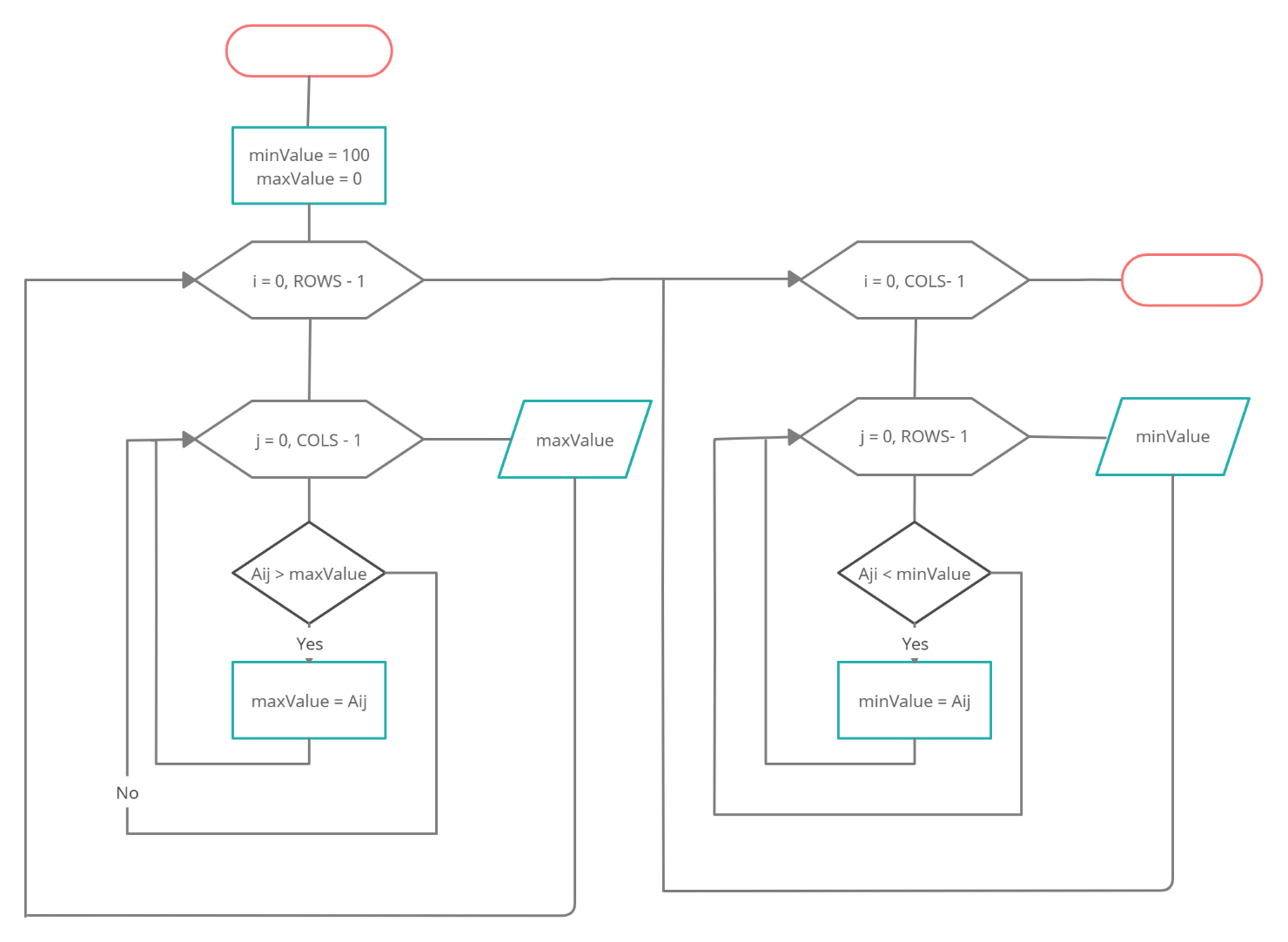


Рисунок 3.1 - Схема алгоритму для задачі 3.

**3.2 Тестування програми**

Для доведення працездатності роботи програми проведемо таку перевірку:

ROWS = 4, COLS = 5 та матрицю:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 7 | 4 | 0 | 9 |
| 4 | 8 | 8 | 2 | 4 |
| 5 | 5 | 1 | 7 | 1 |
| 1 | 5 | 2 | 7 | 6 |

Маємо отримати:

Максимальні значення: 9, 8, 7, 7

Мінімальні значення: 1, 5, 1, 0, 1

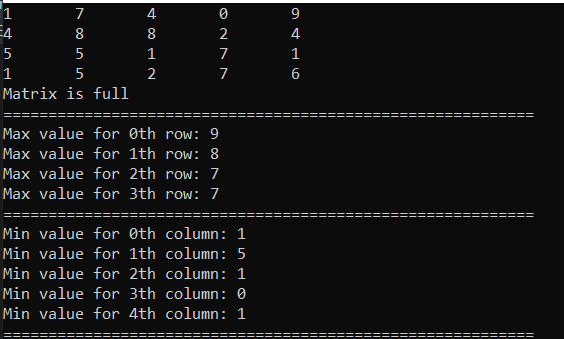


Рисунок 3.2 – Тестування програми.

**3.3 Текст програмною мовою С++.**

#**include** <iostream>

#**define** ROWS 4

#**define** COLS 5

**using namespace** std;

**void** maxValueForRows(int[ROWS][COLS], **int, int, int**);

**void** minValueForColumns(int[ROWS][COLS], **int, int, int**);

**void** main()

{

**int** matrix[ROWS][COLS];

**int** minValue = 100, maxValue = 0;

//Initialize(matrix, ROWS, COLS);

**for** (**int** i = 0; i < ROWS; i++)

{

**for** (**int** j = 0; j < COLS; j++)

{

matrix[i][j] = rand() % 10;

cout << matrix[i][j] << '\t';

}

cout << endl;

}

cout << "Matrix is full" << endl;

cout << "===========================================================" << endl;

maxValueForRows(matrix, ROWS, COLS, maxValue);

cout << "===========================================================" << endl;

minValueForColumns(matrix, ROWS, COLS, minValue);

cout << "===========================================================" << endl;

//deleteMatrixValue(matrix, ROWS, COLS);

}

**void** maxValueForRows(**int** matrix[ROWS][COLS], **int** nRows, **int** nCols, **int** value)

{

**for** (**int** i = 0; i < nRows; i++)

{

**for** (**int** j = 0; j < nCols; j++)

{

**if** (matrix[i][j] > value) value = matrix[i][j];

}

cout << "Max value for " << i << "th row: " << value << endl;

value = 0;

}

}

**void** minValueForColumns(**int** matrix[ROWS][COLS], **int** nRows, **int** nCols, **int** value)

{

**for** (**int** i = 0; i < nCols; i++)

{

**for** (**int** j = 0; j < nRows; j++)

{

**if** (matrix[j][i] < value) value = matrix[j][i];

}

cout << "Min value for " << i << "th column: " << value << endl;

value = 100;

}

}

**Завдання 4**

**1.1 Математична постановка задачі (МПЗ).**

*Вхідні дані:* -

*Вихідні дані:*

count – к-сть рядків, всі елементи яких різні, цілого типу.

*Математична модель задачі:*

Введемо позначення. Константи:

ROWS – кількість рядків матриці.

COLS – кількість стовпців матриці.

Проміжні змінні:

i, j – цілі, змінна для індексів рядка та стовпця.

stop = 0 – ціле, змінна для виходу з циклу з лічильником.

count = ROWS – ціле, к-сть рядків.

Для і = 0, COLS -1:

stop = 0

Для j = 0, ROWS -1, stop = 0 :

Для k = 0, ROWS -1 :

Якщо Aji = Aki:

stop = 1

count = count – 1

Вихід з циклу з лічильником.

Якщо Aji = Aki:

stop = 1

Виведення count

Подання математичної постановки задачі у вигляді таблиці:

Таблиця 4.1

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| Нічого | Для і = 0, COLS -1:  stop = 0  Для j = 0, ROWS -1, stop = 0 :  Для k = 0, ROWS -1 :  Якщо Aji = Aki:  stop = 1  count = count – 1  Вихід з циклу з лічильником.  Якщо Aji = Aki:  stop = 1  Виведення count | count – к-сть рядків, всі елементи яких різні, цілого типу. |
| **Константи:**  ROWS – кількість рядків матриці.  COLS – кількість стовпців матриці.  **Проміжні змінні:**  i, j – цілі, змінна для індексів рядка та стовпця.  stop = 0 – ціле, змінна для виходу з циклу з лічильником.  count = ROWS – ціле, к-сть рядків. |

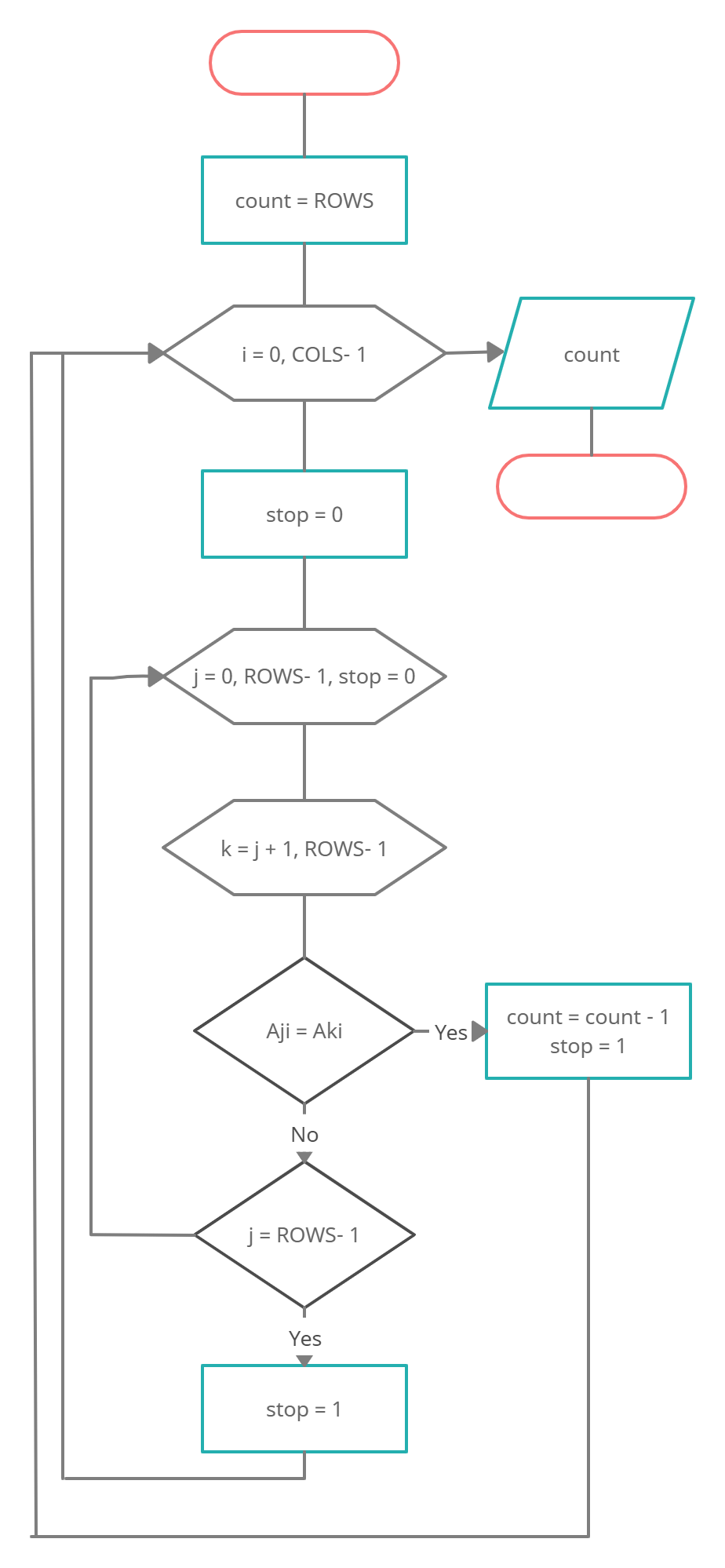


Рисунок 4.1 - Схема алгоритму для задачі 4.

**4.2 Тестування програми**

Для доведення працездатності роботи програми проведемо таку перевірку:

ROWS = 4, COLS = 5 та матрицю:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 5 | 9 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 |
| 5 | 4 | 4 | 3 | 4 |

Маємо отримати: count = 2



Рисунок 4.2 – Тестування програми.

**4.3 Текст програмною мовою С++.**

#**include** <iostream>

#**define** ROWS 4

#**define** COLS 5

**using namespace** std;

**void** main()

{

**int** matrix[ROWS][COLS] = { {1,2,3,4,5},{5,9,6,7,8},

{1,1,1,1,1},{5,4,4,3,4}};

**int** count = ROWS;

**int** stop;

**for** (**int** i = 0; i < ROWS; i++)

{

stop = 0;

**for** (**int** j = 0; j < COLS && !stop; j++)

{

**for** (**int** k = j + 1; k < COLS; k++)

{

//cout << matrix[i][j] << " " << matrix[i][k] << endl;

**if** (matrix[i][j] == matrix[i][k])

{

count--;

stop = 1;

break;

}

**else if** (j == ROWS - 1) stop = 1;

}

}

}

cout << "Count rows with unique items: " << count << endl;

}

**Завдання 5**

**1.1 Математична постановка задачі (МПЗ).**

*Вхідні дані:* -

*Вихідні дані:*

count - к-сть елементів цього стовпчика які більше за його середнє значення, цілого типу.

*Математична модель задачі:*

Введемо позначення. Константи:

ROWS – кількість рядків матриці.

COLS – кількість стовпців матриці.

Проміжні змінні:

i, j – цілі, змінна для індексів рядка та стовпця.

count – дійсне, к-сть елементів у стовпчику більших за його середнє значення. Також використовується як к-сть елементів у стовпчику.

average - дійсне, середнє значення елементів стовпчика.

sum – дійсне, сума елементів стовпчика.

Для і = 0, COLS -1:

sum = 0

count = 0

Для j = 0, ROWS -1:

sum = sum + Aji

count = count + 1

Якщо j + 1 = ROWS:

average = [sum / count]

Для і = 0, ROWS -1:

Якщо Aji > average: count = count + 1

Якщо i + 1 == ROWS:

Виведення count

Вихід з циклу з лічильником

Подання математичної постановки задачі у вигляді таблиці:

Таблиця 5.1

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| Нічого | Для і = 0, COLS -1:  sum = 0  count = 0  Для j = 0, ROWS -1:  sum = sum + Aji  count = count + 1  Якщо j + 1 = ROWS:  average = [sum / count]  Для і = 0, ROWS -1:  Якщо Aji > average: count = count + 1  Якщо i + 1 == ROWS:  Виведення count  Вихід з циклу з лічильником | count - к-сть елементів цього стовпчика які більше за його середнє значення, цілого типу. |
| **Константи:**  ROWS – кількість рядків матриці.  COLS – кількість стовпців матриці.  **Проміжні змінні:**  i, j – цілі, змінна для індексів рядка та стовпця.  count – дійсне, к-сть елементів у стовпчику більших за його середнє значення. Також використовується як к-сть елементів у стовпчику.  average - дійсне, середнє значення елементів стовпчика.  sum – дійсне, сума елементів стовпчика. |

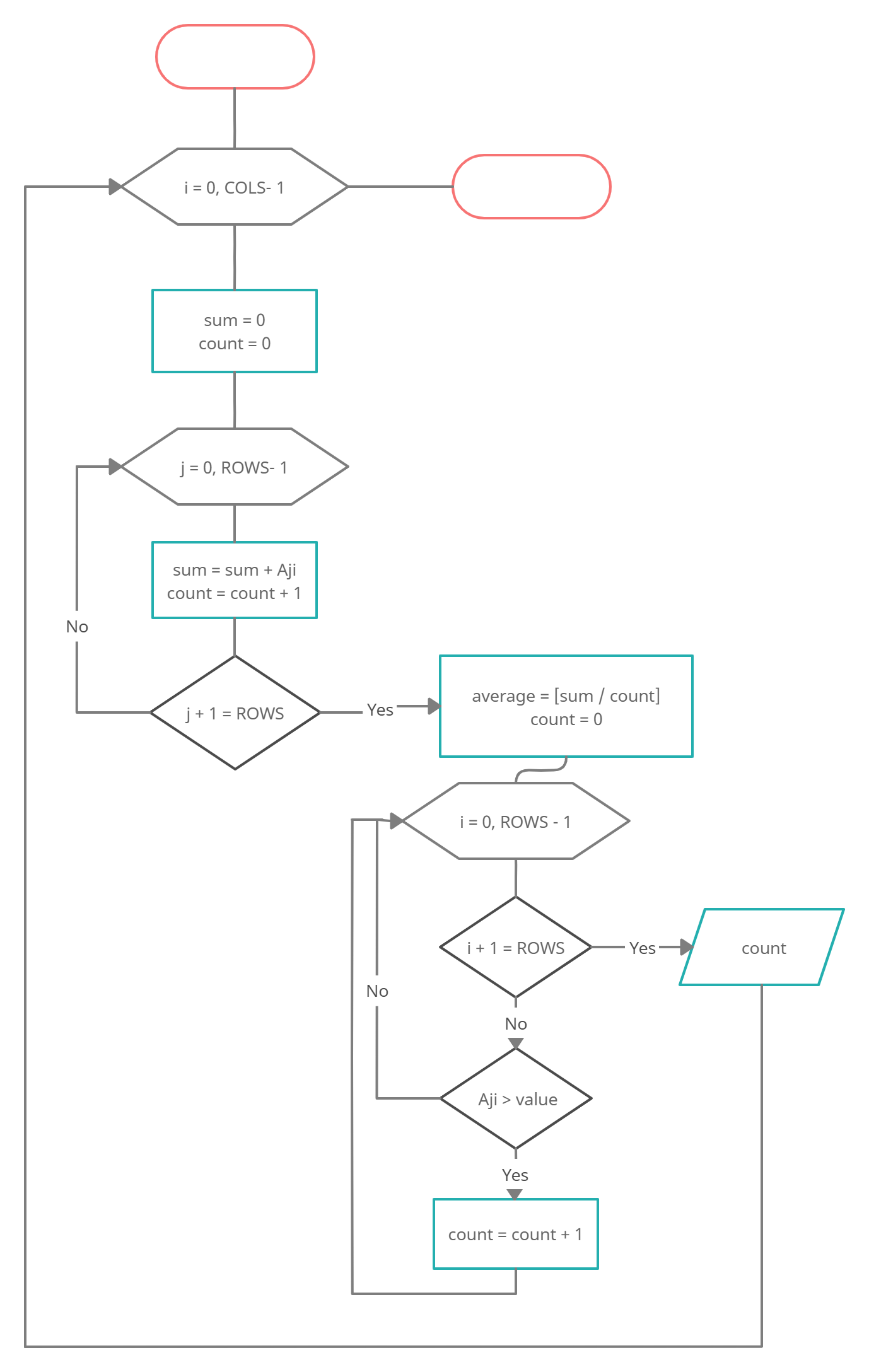


Рисунок 5.1 - Схема алгоритму для задачі 5.

**5.2 Тестування програми**

Для доведення працездатності роботи програми проведемо таку перевірку:

ROWS = 4, COLS = 5 та матрицю:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 5 | 9 | 6 | 7 | 8 |
| 1 | 1 | 1 | 1 | 1 |
| 5 | 4 | 4 | 3 | 4 |

Маємо отримати: 2, 1, 2, 2, 2

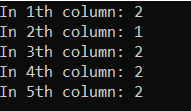
****

Рисунок 5.2 – Тестування програми.

**5.3 Текст програмною мовою С++.**

#**include** <iostream>

#**define** ROWS 4

#**define** COLS 5

**using namespace** std;

**void** Mathcing(**int**[ROWS][COLS], **int, int, double**);

**void** main()

{

**double** sum = 0;

**double** average;

**double** count = 0;

**int** matrix[ROWS][COLS] = { {1,2,3,4,5},{5,9,6,7,8},

{1,1,1,1,1},{5,4,4,3,4} };

**for** (**int** i = 0; i < COLS; i++)

{

**for** (**int** j = 0; j < ROWS; j++)

{

sum += matrix[j][i];

count++;

**if** (j + 1 == ROWS)

{

average = sum / count;

Mathcing(matrix, ROWS, i, average);

}

}

sum = 0;

count = 0;

}

}

**void** Mathcing(**int** matrix[ROWS][COLS], **int** nRows, **int** nCols, **double** value)

{

**int** count = 0;

**for** (**int** i = 0; i < nRows; i++)

{

//cout << matrix[i][nCols] << endl;

**if** (matrix[i][nCols] > value) ++count;

**if** (i + 1 == nRows)

{

cout << "In " << nCols + 1 << "th column: " << count << endl;

}

}

}