

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 7

з дисципліни «Алгоритмізація та програмування»

Тема: «Реалізація алгоритмів обробки двовимірних
масивів на мові C ++»

XAI.301.173.310.02 ЛР

Виконав студент гр. 310

20.02.2024 Софія ПОЛЯКОВА
(підпис, дата) (П.І.Б.)

Перевірив

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

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів (матриць) на мові C ++ і реалізувати оголошення, введення з консолі, обробку і виведення в консоль матриць на мові C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів матриці. Введення і виведення даних здійснити в командному вікні.

Matrix21. Дана матриця розміру $M \times N$. Для кожного рядка матриці з непарним номером (1, 3, ...) знайти середнє арифметичне її елементів. Умовний оператор не використовувати.

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

Matrix69. Дана матриця розміру $M \times N$ і ціле число K ($1 \leq K \leq N$). Після стовпчика матриці з номером K вставити стовпець з одиниць.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення Matrix21.

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

1) m — кількість рядків, цілі числа, $2 \leq m \leq 20$;

2) n — кількість стовпців, цілі числа, $2 \leq n \leq 20$.

Вихідні дані (ім'я, опис, тип):

1) $average$ — середнє арифметичне непарних стовпців, дійсні числа, $average = R$;

2) i — рядок, з якого розраховувалось середнє арифметичне, цілі числа, $i = 0, 2, 4, \dots$ (непарні числа у масиві, який починається з 0).

Алгоритм вирішення показано на рис. 1

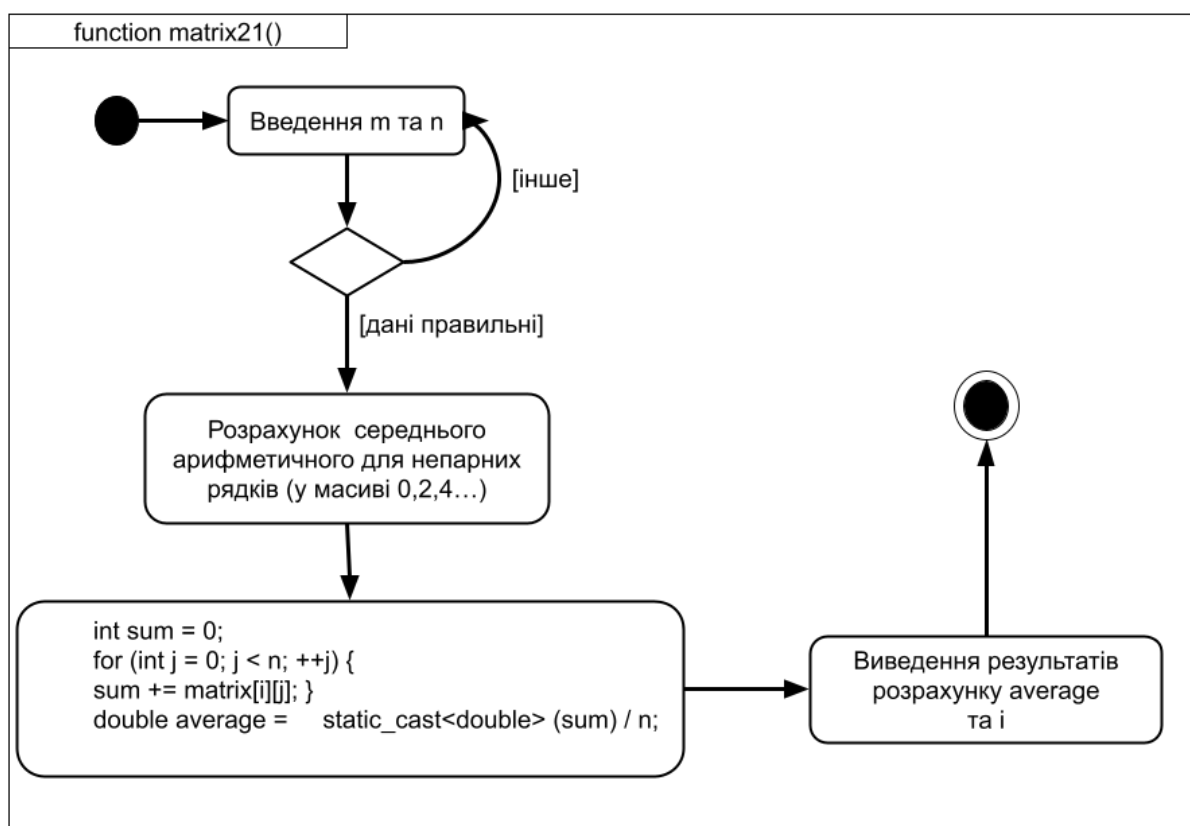


Рисунок 1 – Алгоритм вирішення задачі Matrix21

Лістинг коду вирішення Matrix21 наведено в дод. А (стор. 7).

Екран роботи програми показаний на рис. Б.2, Б.3.

Завдання 2.

Вирішення Matrix69.

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

- 1) m — кількість рядків, цілі числа, $2 \leq m \leq 20$;
- 2) n — кількість стовпців, цілі числа, $2 \leq n \leq 20$;
- 3) k — число для даного стовпця, після якого буде стовпець одиниць, ціле число, $1 < k < n$.

Вихідні дані (ім'я, опис, тип):

$matrix[i][j]$, де i — рядки матриці, дійсні числа;

j — стовпці матриці, дійсні числа.

Алгоритм вирішення показано на рис. 2

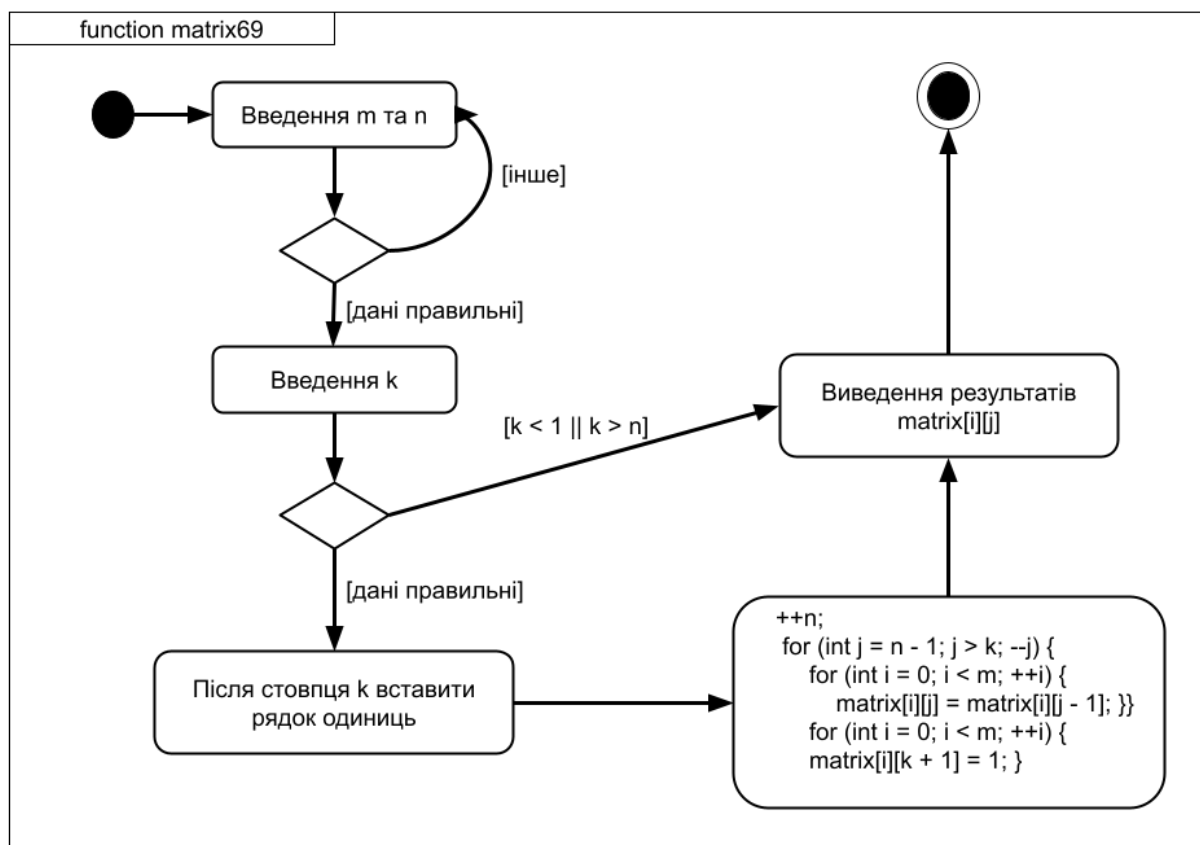


Рисунок 2 – Алгоритм вирішення задачі Matrix69

Лістинг коду вирішення задачі Matrix69 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.1, Б.3 .

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення двовимірних масивів (матриць) на мові C ++. Закріплено на практиці розрахунок та математичні дії з матрицями. Відпрацьовано на коді реалізування оголошення, введення з консолі, обробку і виведення в консоль матриць.

ДОДАТОК А

Лістинг коду програми до задач назви та номери задач

```

#include <iostream>
using namespace std;

const int M = 20;
const int N = 20;
// getting matrixs size
void get_matrix(int matrix[M][N], int& m, int& n);

/* Given a matrix of size M × N. For each row of the matrix with an odd number
(1, 3, ...)
find the arithmetic mean of its elements. Do not use conditional operator.*/
void matrix21(int matrix[M][N], const int& m, const int& n);

/*Given a matrix of size M × N and an integer number K (1 ≤ K ≤ N).
Insert a column of ones after the column with number K in the matrix. */
void matrix69(int matrix[M][N], const int& m, int& n, int k);
void resultMatrix69(int matrix[M][N], int m, int n);

int main() // menu
{
    int task_num; // declaration integer
    do {
        cout << " Enter task number (0 - exit): ";
        cin >> task_num; // input integer
        if (!cin) {
            cout << " Ups!" << endl;
            continue; // error notification
        }
        switch (task_num)
        {
            case 1: { // task1
                int matrix[M][N];
                int m, n;
                get_matrix(matrix, m, n);
                matrix21(matrix, m, n);
                break;
            }
            case 2: { // task2
                int matrix[M][N];
                int m, n, k;
                get_matrix(matrix, m, n);
                cout << " Matrix 69 " << endl;
                cout << "Enter the column number where to insert 1s : ";
                cin >> k;
                matrix69(matrix, m, n, k);
                resultMatrix69(matrix, m, n);
            }
        }
    } while (task_num != 0);
}

```

```

        break;
    }
    case 0:
        cout << " Bye!" << endl;
        break; // exit from the program
    default:
        cout << " Wrong task number!" << endl;
        // output for incorrect numbers
        break;
    }
} while (task_num != 0); // end the program
return 0;
}

// matrix_size
void get_matrix(int matrix[M][N], int& m, int& n) {
    do {
        // Input the size of the matrix
        cout << "Enter the number of rows (2-20) (M): ";
        cin >> m;
        cout << "Enter the number of columns (= Rows !!!) (2-20) (N): ";
        cin >> n;
    } while (n < 2 || n > N || m < 2 || m > M || m != n);

    // Input the elements of the matrix
    cout << "Enter the matrix elements: " << endl;
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            cout << "Element [" << i << "][" << j << "]: ";
            cin >> matrix[i][j];
        }
    }
}

// task1
void matrix21(int matrix[M][N], const int& m, const int& n) {
    // Output the arithmetic mean for odd-numbered rows
    cout << " Matrix 21 " << endl;
    cout << "Arithmetic mean for odd-numbered rows: " << endl;
    for (int i = 0; i < m; i += 2) {
        int sum = 0;
        for (int j = 0; j < n; ++j) {
            sum += matrix[i][j];
        }
        double average = static_cast<double>(sum) / n;
        cout << "Row " << i << ": " << average << endl;
    }
}

// task2

```

```

void matrix69(int matrix[M][N], const int& m, int& n, int k) {
    if (k < 1 || k > n) {
        cout << "Invalid column number!" << endl;
        return;
    }

    // Check if the new number of columns exceeds N
    if (n + 1 > N) {
        cout << "Cannot insert column, matrix is full!" << endl;
        return;
    }

    // Increase the number of columns by 1
    ++n;

    // Shift all columns to the right of k by one position to the right
    for (int j = n - 1; j > k; --j) {
        for (int i = 0; i < m; ++i) {
            matrix[i][j] = matrix[i][j - 1];
        }
    }

    // Insert a column with 1s at position k
    for (int i = 0; i < m; ++i) {
        matrix[i][k + 1] = 1;
    }
}

// results task2
void resultMatrix69(int matrix[M][N], int m, int n) {
    cout << "Resulting matrix:" << endl;
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            cout << " " << matrix[i][j];
        }
        cout << endl;
    }
}

```


ДОДАТОК Б

Скрін-шоти вікна виконання програми

```

Enter task number (0 - exit): 1
Enter the number of rows (2-20) (M): 4
Enter the number of columns (= Rows !!!) (2-20) (N): 4
Enter the matrix elements:
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [0][3]: 4
Element [1][0]: 5
Element [1][1]: -6
Element [1][2]: -7
Element [1][3]: -8
Element [2][0]: -12
Element [2][1]: 13
Element [2][2]: 15
Element [2][3]: 90
Element [3][0]: 14
Element [3][1]: 29
Element [3][2]: 16
Element [3][3]: 13
Matrix 21
Arithmetic mean for odd-numbered rows:
Row 0: 2.5
Row 2: 26.5
Enter task number (0 - exit): 0
Bye!

```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Matrix21

```

Enter task number (0 - exit): 2
Enter the number of rows (2-20) (M): 3
Enter the number of columns (= Rows !!!) (2-20) (N): 3
Enter the matrix elements:
Element [0][0]: -2
Element [0][1]: 33
Element [0][2]: 45
Element [1][0]: 0
Element [1][1]: 12
Element [1][2]: 35
Element [2][0]: 64
Element [2][1]: -67
Element [2][2]: 2
Matrix 69
Enter the column number where to insert 1s : 2
Resulting matrix:
-2 33 45 1
0 12 35 1
64 -67 2 1

```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Matrix69

```
Enter task number (0 - exit): 1
Enter the number of rows (2-20) (M): 3
Enter the number of columns (= Rows !!!) (2-20) (N): 3
Enter the matrix elements:
Element [0][0]: -1
Element [0][1]: 2
Element [0][2]: 3
Element [1][0]: -4
Element [1][1]: 5
Element [1][2]: -6
Element [2][0]: -7
Element [2][1]: -8
Element [2][2]: 9
Matrix 21
Arithmetic mean for odd-numbered rows:
Row 0: 1.33333
Row 2: -2
Enter task number (0 - exit): 2
Enter the number of rows (2-20) (M): 3
Enter the number of columns (= Rows !!!) (2-20) (N): 3
Enter the matrix elements:
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [1][0]: 4
Element [1][1]: 5
Element [1][2]: 6
Element [2][0]: 7
Element [2][1]: 8
Element [2][2]: 9
Matrix 69
Enter the column number where to insert 1s : 1
Resulting matrix:
1 2 1 3
4 5 1 6
7 8 1 9
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання
Matrix21 та Matrix69