

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів

Кафедра систем управління літальних апаратів

Лабораторна робота №7

з дисципліни «Алгоритмізація та програмування»

Тема: "Реалізація циклічних алгоритмів мовою C++"

XAI.301.175.318.20 ЛР

Виконав студент гр. 318

Аліна ХОБОТ

(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) у мові C ++ і реалізувати декларацію, введення з консолі, обробку і виведення в консоль матриць мовою 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.

Matrix30. Дана матриця розміру $M \times N$. У кожному її стовпці знайти кількість елементів, більших ніж середнє арифметичне всіх елементів цього стовпчика.

Рис. 1 – Умова завдання таблиця 1, варіант Matrix30.

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

Matrix81. Дана квадратна матриця порядку M . Обнулити елементи матриці, що лежать одночасно вище головної діагоналі і нижче побічної діагоналі. Умовний оператор не використовувати.

Рис. 2 – Умова завдання таблиця 2, варіант Matrix81.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення завдання таблиця 1, варіант Matrix30.

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

int M - кількість рядків матриці, $2 \leq M \leq 20$.

int N - кількість стовпців матриці, $2 \leq N \leq 20$.

int matrix - матриця елементів, від -10^6 до 10^6 .

Вихідні дані (ім'я, опис, тип):

countAboveAverage - кількість елементів у кожному стовпці, більших за середнє арифметичне, int[] (масив розміром N).

Текстове повідомлення - виведення кількості для кожного стовпця, string.

Лістинг коду вирішення завдання таблиця 1, варіант Matrix30 наведено в дод. А (стор. 4).

Екран роботи програми показаний на рис. Б.1 (дод. Б, стор. 8)

Завдання 2.

Вирішення завдання таблиця 2, варіант Matrix81.

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

int M - кількість рядків матриці, $2 \leq M \leq 20$.

int matrix - матриця елементів, від -10^6 до 10^6 .

Вихідні дані (ім'я, опис, тип):

int matrix – оновлена матриця елементів.

Текстове повідомлення - виведення матриці після обнулення, string.

Лістинг коду вирішення завдання таблиця 2, варіант Matrix81 наведено в дод. А (стор. 4).

Екран роботи програми показаний на рис. Б.2 (дод. Б, стор. 9)

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення двовимірних масивів (матриць) у мові C++ і реалізуваї декларацію, введення з консолі, обробку і виведення в консоль матриць мовою C++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#include <iomanip>
using namespace std;

// Функція для введення розмірів матриці (M і N)
void inputSize(int &M, int &N) {
    cout << "Введіть кількість рядків матриці (2-20): ";
    cin >> M;
    // Перевірка на допустимий діапазон
    while (M < 2 || M > 20) {
        cout << "Некоректне значення! Введіть ще раз (2-20): ";
        cin >> M;
    }

    cout << "Введіть кількість стовпців матриці (2-20): ";
    cin >> N;
    // Перевірка на допустимий діапазон
    while (N < 2 || N > 20) {
        cout << "Некоректне значення! Введіть ще раз (2-20): ";
        cin >> N;
    }
}

// Функція для введення розміру квадратної матриці (тільки M)
void inputSquareSize(int &M) {
    cout << "Введіть порядок квадратної матриці (2-20): ";
    cin >> M;
    // Перевірка на допустимий діапазон
    while (M < 2 || M > 20) {
        cout << "Некоректне значення! Введіть ще раз (2-20): ";
        cin >> M;
    }
}

// Функція для введення елементів матриці
void inputMatrix(int matrix[][20], int M, int N) {
    cout << "Введіть елементи матриці:\n";
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < N; ++j) {
            cin >> matrix[i][j];
            // Перевірка діапазону значення елемента
            while (matrix[i][j] < -1000000 || matrix[i][j] > 1000000) {
                cout << "Некоректне значення! Введіть ще раз: ";
                cin >> matrix[i][j];
            }
        }
    }
}
```

```

    }
}

// Функція для виведення матриці
void outputMatrix(int matrix[][20], int M, int N) {
    cout << "Матриця:\n";
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < N; ++j) {
            // Форматований вивід кожного елемента з вирівнюванням
            cout << setw(8) << matrix[i][j];
        }
        cout << endl;
    }
}

// Функція для обробки Matrix30
// Обчислює середнє арифметичне кожного стовпця та рахує кількість елементів,
// більших за це середнє
void processMatrix30(int matrix[][20], int M, int N, int countAboveAverage[]) {
    for (int j = 0; j < N; ++j) {
        double sum = 0;

        // Сума всіх елементів стовпця
        for (int i = 0; i < M; ++i) {
            sum += matrix[i][j];
        }

        double average = sum / M;
        int count = 0;

        // Підрахунок кількості елементів у стовпці більші за середнє
        for (int i = 0; i < M; ++i) {
            if (matrix[i][j] > average) {
                count++;
            }
        }
        countAboveAverage[j] = count;
    }
}

// Функція для виведення результату Matrix30
// Виводить для кожного стовпця кількість елементів, які більші за середнє
// арифметичне
void outputResultMatrix30(int countAboveAverage[], int N) {
    cout << "\nКількість елементів у кожному стовпці, більших за середнє
арифметичне:\n";
    for (int j = 0; j < N; ++j) {
        cout << "Стовпець " << j + 1 << ": " << countAboveAverage[j] << endl;
    }
}

```

```

}

// Функція для обробки Matrix81
// Обнуляє всі елементи вище головної діагоналі (i < j)
// та нижче побічної діагоналі (i > M - j - 1)
void processMatrix81(int matrix[][20], int M) {
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < M; ++j) {
            // Обнуляємо, якщо елемент або вище головної, або нижче побічної
            // діагоналі
            matrix[i][j] *= !(i < j || i > M - j - 1);
        }
    }
}

// Основна функція для Matrix30
void taskMatrix30() {
    int matrix[20][20]; // Матриця для зберігання даних
    int countAboveAverage[20]; // Масив для зберігання результатів
    int M, N; // Розміри матриці

    inputSize(M, N); // Введення розмірів
    inputMatrix(matrix, M, N); // Введення елементів
    outputMatrix(matrix, M, N); // Виведення початкової матриці
    processMatrix30(matrix, M, N, countAboveAverage); // Обробка
    outputResultMatrix30(countAboveAverage, N); // Виведення результатів
}

// Основна функція для Matrix81
void taskMatrix81() {
    int matrix[20][20]; // Квадратна матриця
    int M; // Розмір квадратної матриці

    inputSquareSize(M); // Введення порядку матриці
    inputMatrix(matrix, M, M); // Введення елементів
    outputMatrix(matrix, M, M); // Виведення початкової матриці
    processMatrix81(matrix, M); // Обробка – обнулення елементів
    cout << "\nМатриця після обнулення елементів:\n";
    outputMatrix(matrix, M, M); // Виведення результату
}

// Функція для багаторазового вибору завдання
void mainMenu() {
    int choice;
    do {
        // Меню
        cout << "\nОберіть завдання:\n";
        cout << "1 - Виконати Завдання1\n"; // Matrix30
        cout << "2 - Виконати Завдання2\n"; // Matrix81
        cout << "0 - Вийти\n";
    } while (choice != 0);
}

```

```
cout << "Ваш вибір: ";
cin >> choice;

// Обробка вибору
switch (choice) {
    case 1:
        taskMatrix30(); // Виконання першого завдання
        break;
    case 2:
        taskMatrix81(); // Виконання другого завдання
        break;
    case 0:
        cout << "Завершення програми.\n";
        break;
    default:
        cout << "Некоректний вибір! Спробуйте ще раз.\n";
}
} while (choice != 0); // Повторювати, поки не обрано вихід
}

// Точка входу в програму
int main() {
    mainMenu(); // Виклик головного меню
    return 0;
}
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```

Оберіть завдання:
1 - Виконати Завдання1
2 - Виконати Завдання2
0 - Вийти
Ваш вибір: 1
Введіть кількість рядків матриці (2-20): 4
Введіть кількість стовпців матриці (2-20): 4
Введіть елементи матриці:
3
6
8
6
5
4
5
2
4
5
6
7
8
9
6
7
Матриця:
      3      6      8      6
      5      4      5      2
      4      5      6      7
      8      9      6      7

Кількість елементів у кожному стовпці більших за середнє арифметичне:
Стовпець 1: 1
Стовпець 2: 1
Стовпець 3: 1
Стовпець 4: 3

```

Рисунок Б.1 – Екран виконання програми для вирішення завдання таблиця 1,
варіант Matrix30.


```

Оберіть завдання:
1 - Виконати Завдання1
2 - Виконати Завдання2
0 - Вийти
Ваш вибір: 2
Введіть порядок квадратної матриці (2-20): 5
Введіть елементи матриці:
3
2
3
3
3
32
2
2
2
2
2
24
4
5
54
4
4
4
4
3
3
3
3
3
Матриця:
      3      2      3      3      3
    32      2      2      2      2
      2      2     24      4      5
    54      4      4      4      4
      3      3      3      3      3

Матриця після обнулення елементів:
Матриця:
      3      0      0      0      0
    32      2      0      0      0
      2      2     24      0      0
    54      4      0      0      0
      3      0      0      0      0

Оберіть завдання:
1 - Виконати Завдання1
2 - Виконати Завдання2
0 - Вийти
Ваш вибір: 0
Завершення програми.

...Program finished with exit code 0

```

Рисунок Б.2 – Екран виконання програми для вирішення завдання таблиця 2, варіант Matrix81.



Рисунок Б.3 – Діаграма для завдання таблиця 1, варіант Matrix30.



Рисунок Б.4 – Діаграма для завдання таблиця 2, варіант Matrix81.