Фаховий коледж ракетно-космічного машинобудування

Дніпровського національного університету імені Олеся Гончара

ЗВІТ

з лабораторних робіт

з дисципліни «Технології (Програмування)»

Спеціальність 123 Комп’ютерна інженерія

Група КС-21-1

Виконав А.С. Василенко

Перевірив М.М. Гапоненко

2022-2023

ЗМІСТ

Лабораторна робота № 17

ЛАБОРАТОРНА РОБОТА № 17

Тема: Складання та налагодження лінійних програм.

Мета: Придбати практичні навички при написанні та налагодженні програм лінійних алгоритмів на мові програмування Сі\С++.

Хід роботи

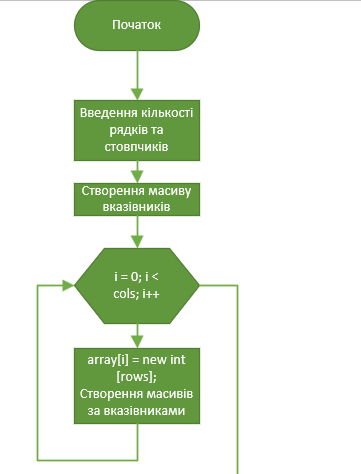
1 Постановка задачі.

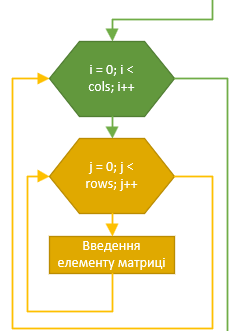
Загальна постановка завдання:

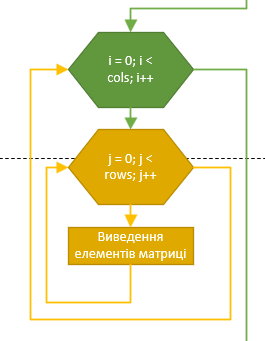
Розмірність матриці (кількість рядків та стовпчиків) користувач вводить з клавіатури. Створити відповідну динамічну матрицю. Обробити її за алгоритмом згідно Вашого варіанта. Програма повинна задовольняти наступним вимогам:

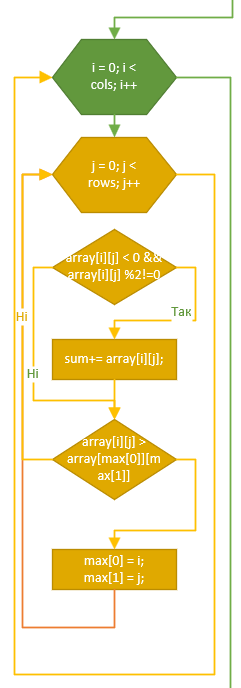
* введення значень елементів матриці проводити в окремому циклі з відповідними коментарями;
* вивести введену матрицю в загальноприйнятому вигляді;
* при обробці матриці враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
* введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.
* Знайти максимальний та обчислити кількість від’ємних непарних елементів у кожному з рядків матриці.

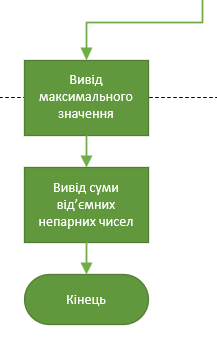
2 Блок-схема алгоритму рішення задачі.











3 Текст програми мовою програмування С++.

#include <iostream> // для вводу та виводу

#include <stdio.h> // для printf, не обов'язково, але так гарніше виводиться масив

#include <windows.h> // для встановлення кодування

using namespace std;

int main() {

    SetConsoleCP(65001); // встановлення кодування

    SetConsoleOutputCP(65001); // встановлення кодування

    int rows, cols, sum=0; // рядки, стовпчики, та змінна для суми від'ємних непарних елементів

    int max[] = {0,0}; // масив для зберігання індексів максимального числа у матиці

    cout << "Введіть кількість рядків та стовпчиків: " << endl;

    cin >> rows >> cols;

    // Вивільнення місця під масив вказівників

    int\*\*array = new int\*[cols];

    for (int i = 0; i < cols; i++) {

        // створення динамічного масиву елементів у кожному з вказівників

        array[i] = new int [rows];

    }

    // Введення елементів у матрицю

    cout << "Введіть матрицю: " << endl;

    for (int i = 0; i < cols; i++) {

        for (int j = 0; j < rows; j++) {

            cout << "Введіть ["<<i<<","<<j<<"] елемент: ";

            cin >> array[i][j];

        }

    }

    // Вивід матриці

    cout << "Матриця:" << endl;

    for (int i = 0; i < cols; i++) {

        cout << endl;

        for (int j = 0; j < rows; j++) {

            printf("%3d ", array[i][j]);

            // printf потрібен для виведення чисел з резервацією місця 3

            // тобто замість:3 100 5

            //          буде:  3 100   5

            // це потрібно, аби масив виводився рівно:

            //  21  32  -5

            //  -5 -10   3

        }

    }

    // Пошук максимального числа у матриці, а також сума від'ємних непарних елементів

    for (int i = 0; i < cols; i++) {

        for (int j = 0; j < rows; j++) {

            if (array[i][j] < 0 && array[i][j] %2!= 0)

                sum += array[i][j];

            // оновлення координат максимального елемнту

            if (array[i][j] > array[max[0]][max[1]]) {

                max[0] = i;

                max[1] = j;

            }

        }

    }

    cout << endl << "Найбільше значення: " << array[max[0]][max[1]] << endl;

    cout << "Сума від'ємних непарних елементів: " << sum << endl;

    // Вивільнення матриці

    // Спочатку ми вивільняжмо масиви

    for (int i = 0; i < cols; i++) {

        delete[] array[i];

    }

    // Потім ми вивільняємо масив із вказівниками які раніше вказували на масиви

    delete[] array;

    // І потім видаляємо адресу вказівника, так як вона нам більше не потрібна

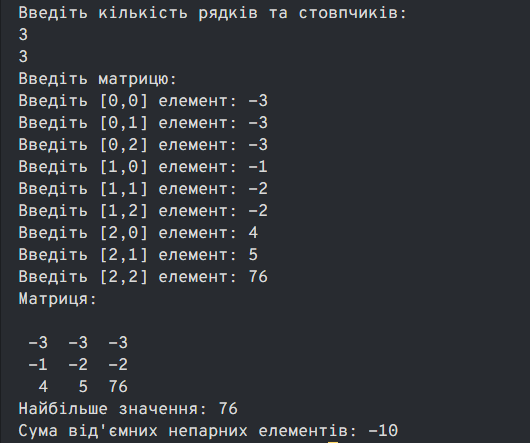
    array = NULL;

    // Якщо робити це у іншому порядку, то ми не зможемо вивільнити всю зайняту пам'ять, так як втратимо вказівники на елементи

    return 0;

}

4 Копія вікна виконання програми



5 Висновок

В результаті виконання лабораторної роботи опрацьовано теоретичний матеріал за темами:

* Динамічні матриці