«Отчет по ПЗ 1.2. Динамические массивы (матрицы)» Никита Преображенский \*\*\*. Вариант 30.

1. Задание: все положительные элементы матрицы заменить значением элемента, стоящего в ее последней строке и последнем столбце.
2. Постановка задачи.

Входные данные + ограничения:

А) Переменная N. Тип int. Размерность матрицы (количество строк).

Исключения: буква, строка, вещественное, меньше или равно 0, большие числа (зависит от наличия свободной оперативной памяти компьютера).

Б) Переменная M. Тип int. Размерность матрицы (количество столбцов).

Исключения: буква, строка, вещественное, меньше или равно 0, большие числа (зависит от наличия свободной оперативной памяти компьютера).

Г) Переменная matrix[][]. Тип \*\*double. Элементы матрицы.

Исключения: буква, строка.

Выходные данные + ограничения:

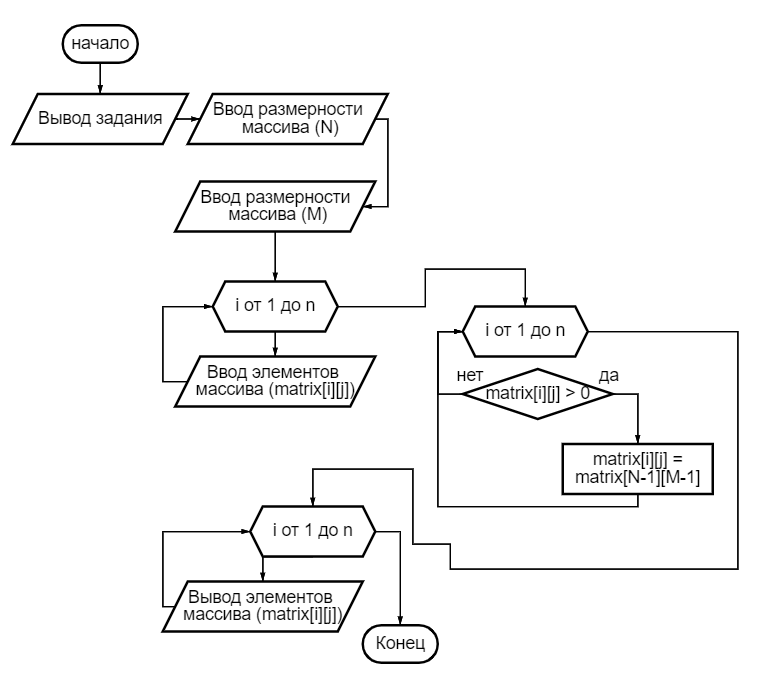
А) Переменная matrix[][]. Тип \*\*double. Элементы матрицы после обработки.

Исключения: буква, буква с цифрой, строка.

Среда разработки: Visual Studio 2022.

2. Структура программы.

2.1 Блок-схема алгоритма



3. Тесты.

|  |  |  |
| --- | --- | --- |
| Входные данные | Результат программы | Смысл |
| n = a | Неверное значение длины массива! | Проверка на букву |
| n = 3,2 | Неверное значение длины массива! | Проверка на целое число |
| n = 10000000000000000 | Неверное значение длины массива! | Выделение памяти массивам |
| n = 0 | Неверное значение длины массива! | Валидность размера массива |
| n = zxcgul | Неверное значение длины массива! | Проверка на строку |
| m = b | Неверное значение длины массива! | Проверка на букву |
| m = 4a | Неверное значение длины массива! | Проверка на число c буквой |
| m = 10000000000000000 | Неверное значение длины массива! | Выделение памяти массивам |
| m = 0 | Неверное значение длины массива! | Валидность размера массива |
| m = zxcgul | Неверное значение длины массива! | Проверка на строку |
| matrix[i][j] = a | Неверное значение элемента массива! | Проверка на букву |
| matrix[i][j] = asadsa | Неверное значение элемента массива! | Проверка на строку |
| N = 2  M = 2  matrix[0][0] = -1  matrix[0][1] = 5  matrix[1][0] = 6  matrix[1][1] = 0 | Ваши элементы массива:  Элемент [0][0] = -1  Элемент [0][1] = 0  Элемент [1][0] = 0  Элемент [1][1] = 0 | Проверка на корректность работы программы c разными по знаку числами |
| N = 2  M = 2  matrix[0][0] = -1  matrix[0][1] = -2  matrix[1][0] = -2.5  matrix[1][1] = -6 | Ваши элементы массива: Элемент [0][0] = -1  Элемент [0][1] = -2  Элемент [1][0] = -2.5  Элемент [1][1] = -6 | Проверка на корректность работы программы без положительных чисел |

4. Код:

#include <iostream> // Подключение стандартной библиотеки

using namespace std; // Использование стандартного пространства имён

void InputDouble(double\*); // Прототип функции ввода элементов матрицы

void InputInt(int\*); // Прототип функции ввода размерности матрицы

int main(void)

{

setlocale(LC\_ALL, "RUS"); // Установка русского языка в консоль

cout << "\t\t№ 30. Все положительные элементы матрицы заменить значением элемента," << endl;

cout << "\t\tстоящего в ее последней строке и последнем столбце.\n\n";

int N = 0, M = 0;

cout << "\tВведите количество строк массива:\n";

InputInt(&N); // Ввод размерности матрицы

cout << "\tВведите количество столбцов массива:\n";

InputInt(&M); // Ввод размерности матрицы

double\*\* matrix;

matrix = new double\* [N];

for (int i = 0; i < N; i++) // Выделение памяти под матрицу

{

matrix[i] = new double[M];

}

cout << "\tВведите элементы массива:\n";

for (int i = 0; i < N; i++) // Ввод элементов

{

for (int j = 0; j < M; j++)

{

cout << "\t\tЭлемент" << " [" << i << "][" << j << "]\n";

InputDouble(&matrix[i][j]);

}

}

for (int i = 0; i < N; i++) // Обработка элементов

{

for (int j = 0; j < M; j++)

{

if (matrix[i][j] > 0)

{

matrix[i][j] = matrix[N-1][M-1];

}

}

}

cout << "\tВаши элементы массива:\n";

for (int i = 0; i < N; i++) // Вывод обработанных элементов

{

for (int j = 0; j < M; j++)

{

cout << "\t\tЭлемент" << " [" << i << "][" << j << "] = " << matrix[i][j] << endl;

}

}

for (int i = 0; i < N; i++) // Освобождение памяти из под матрицу

{

free(matrix[i]);

}

free(matrix);

cout << endl;

return 0;

}

void InputInt(int\* tmp) // Описание функции ввода размеров матрицы

{

while (!(cin >> \*tmp) || (cin.peek() != '\n') || (\*tmp <= 0))

{

cin.clear();

while (cin.get() != '\n');

cout << "\tIncorrect input!" << endl;

}

}

void InputDouble(double\* tmp) // Описание функции ввода элементов матрицы

{

while (!(cin >> \*tmp) || (cin.peek() != '\n'))

{

cin.clear();

while (cin.get() != '\n');

cout << "\tIncorrect input!" << endl;

}

}