«Отчет по ПЗ 1.2. Динамические массивы (матрицы)» Асадов Нихат \*\*\*. Вариант 2.

Задание: Заменить элементы матрицы, равные нулю, на заданное значение.

1. Постановка задачи.

Входные данные + ограничения:

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

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

В) Переменная value. Тип float. Значение, которое будет заменять нулевые элементы матрицы.

Исключения: буква, буква с цифрой, строка, большие числа.

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

Исключения: буква, буква с цифрой, строка.

Выходные данные + ограничения:

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

Исключения: буква, буква с цифрой, строка.

Промежуточные данные + ограничения:

А) Переменная flag. Тип int. Флаг переменная для сигнализирования наличия ошибок.

Исключения: буква, буква с цифрой, строка, вещественное число, числа не равные 0 или 1.

Б) Переменная n. Тип float. Промежуточная переменная для проверки на буквенное значение размера массива.

Исключения: буква, буква с цифрой, строка, значение меньше 1.

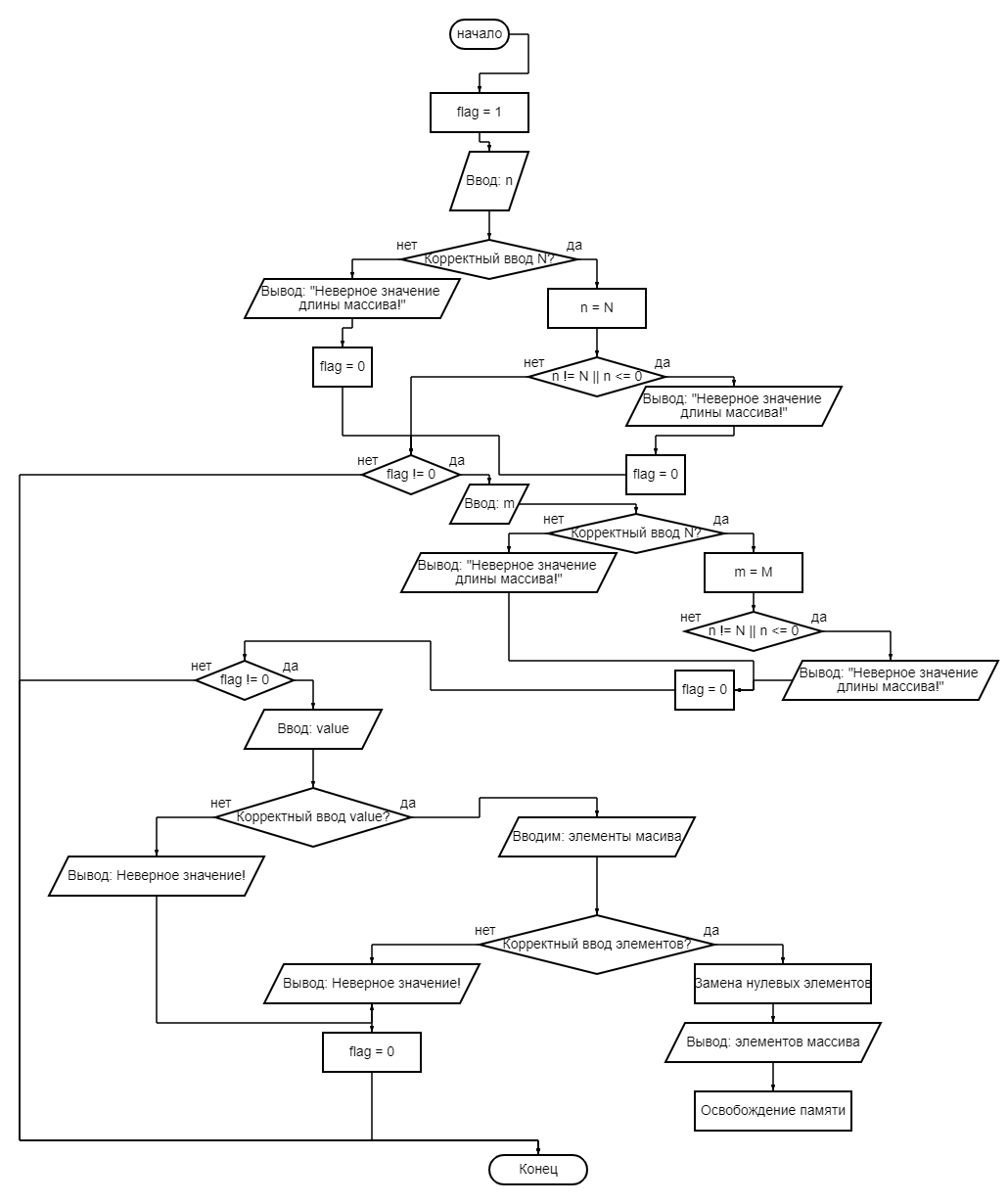
В) Переменная a. Тип char. Промежуточная переменная для проверки на отсутствие буквенных значений в размерах массивов

Исключения: нет.

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 = 3a | Неверное значение длины массива! | Проверка на число c буквой |
| 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 = 3,2 | Неверное значение длины массива! | Проверка на целое число |
| 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 |
| value = a | Неверное значение! | Проверка на букву |
| value = 1a | Неверное значение! | Проверка на число c буквой |
| value = adsada | Неверное значение! | Проверка на строку |
| matrix[i][j] = a | Неверное значение элемента массива! | Проверка на букву |
| matrix[i][j] = 666rain | Неверное значение элемента массива! | Проверка на число c буквой |
| matrix[i][j] = asadsa | Неверное значение элемента массива! | Проверка на строку |
| n = 2  m = 2  value = 6  matrix[0][0] = 2,5  matrix[0][1] = 5,2  matrix[1][0] = 6  matrix[1][1] = 5 | Элементы вашего массива:  Элемента массива [0][0] = 2.5  Элемента массива [0][1] = 5.2  Элемента массива [1][0] = 6  Элемента массива [1][1] = 4 | Проверка на корректность работы программы без замены нулевого элемента |
| n = 2  m = 2  value = 6  matrix[0][0] = 0  matrix[0][1] = 5,2  matrix[1][0] = 6  matrix[1][1] = 5 | Элементы вашего массива:  Элемента массива [0][0] = 6  Элемента массива [0][1] = 5.2  Элемента массива [1][0] = 6  Элемента массива [1][1] = 5 | Проверка на корректность работы программы c заменой нулевого элемента |

4. Код:

// ConsoleApplication26.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

#define \_CRT\_SECURE\_NO\_WARNINGS // Директива препроцесса для устранения предупреждений

#include <iostream>

#include <cmath>

#include <stdio.h>

using namespace std; // Использование стандартного пространства имён

int main()

{

setlocale(LC\_ALL, "RUS"); // Установка для консоли русского языка

cout << "Каждая пара (Xk ,Yk) представляет координаты одной из n точек на плоскости.";

cout << "Определить, у какого числа точек\nположительна как абсцисса Xk, так и ордината Yk; для прочих точек найти среднее расстояние до начала координат\n\n";

int n; // Размер массивов

float N; // Промежуточная переменная для проверки на целостность размеров массивов

char a; // Промежуточная переменная для проверки на отсутствие буквенных значений в размерах массивов

cout << "Введите размер массивов:\n";

if (scanf("%f%c", &N, &a) != 2 || a != '\n') // Проверка размеров массивов на буквенное значение

{

cout << "Неверное значение длины массивов!";

}

else

{

n = N;

if (n != N || n <= 0) // Проверка на целостность и положительность размера массивов

{

cout << "Неверное значение длины массивов!";

}

else

{

float\* X = new float[n]; // Создание массива X на n элементов

float\* Y = new float[n]; // Создание массива Y на n элементов

float num; // Промежуточное значение

int flag = 1; // Промежуточная переменная для проверки на валидность элемента

cout << endl;

cout << "Введите " << n << " пар-(ы) точек:\n";

for (int i = 0; i < n; i++) // Заполнение массивов

{

if (scanf("%f", &num) != 1) // Проверка координаты n-ой точки на букву

{

cout << "Неверное значение для элемента массива!";

flag = 0; // Переключение флага для избегания дальнейших действий

break; // Преждевременный выход из цикла из-за некорректности ввода координат n-ой точки

}

else

{

X[i] = num; // Присваиваем X координате n-ой точки введённое значение

}

if (i == (n - 1)) // Проверка на нахождении на последнем элементе массива

{

if (scanf("%f%c", &num, &a) != 2 || a != '\n') // Проверка значения элемента массива на буквенное значение

{

cout << "Неверное значение для элемента массива!";

flag = 0; // Переключение флага для избегания дальнейших действий

break; // Преждевременный выход из цикла из-за некорректности ввода координат n-ой точки

}

else

{

Y[i] = num; // Присваиваем Y координате n-ой точки введённое значение

}

}

else

{

if (scanf("%f", &num) != 1) // Проверка введённого элемента массива на корректность ввода

{

cout << "Неверное значение для элемента массива!"; // Проверка элемента массива на буквенное значение

flag = 0; // Переключение флага для избегания дальнейших действий

break; // Преждевременный выход из цикла из-за некорректности ввода координат n-ой точки

}

else

{

Y[i] = num; // Присваиваем Y координате n-ой точки введённое значение

}

}

}

if (flag == 1) // Проверка флага на наличие ошибок

{

int count = 0; // Сумма положительных точек

float rast = 0; // Сумма растояния прочих точек

int schet = 0; // Счетчик прочих точек

for (int i = 0; i < n; i++) // Основные действия алгоритма

{

if (X[i] > 0 && Y[i] > 0) // Поиск положительных элементов массивов

{

count++; // Увеличиваем счётчик подходящих переменнхы

}

else

{

rast += sqrt(pow(fabs(X[i]), 2) + pow(fabs(Y[i]), 2)); // Прибавляем к сумме растояний до начала координат

schet++; // Увеличиваем кол-во прочих точек

}

}

cout << endl;

cout << "Кол-во точек с положительной абсциссой и ординатой: " << count << endl;

if (schet == 0) // Проверка на наличие прочих точек

{

cout << endl;

cout << "Прочих точек нет.";

}

else

{

float sred = rast / schet; // Расчёт среднего расстояния до начала координат

cout << endl;

cout << "Среднее расстояние до начала координат прочих точек: " << sred;

}

}

cout << "\n\n\n";

delete(X); // Очистка памяти

delete(Y); // Очистка памяти

}

}

return 0;

}