**ЛАБОРАТОРНАЯ РАБОТА №6**

**«ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ С ПОМОЩЬЮ ФУНКЦИЙ»**

**1. Цель работы**

Изучить основные принципы обработки двумерных массивов, исследовать способы передачи параметров в функции.

**2. Вариант задания**

Определить количество и сумму нечетных отрицательных элементов.

Соседями элемента *aij* в матрице *A* назовем элементы *аkl*, для которых верно следующее: *i-1 ≤ k ≤ i+1*, *j-1 ≤ l ≤ j+1*. Элемент матрицы называется локальным максимумом, если он строго больше всех имеющихся у него соседей. Подсчитать количество локальных максимумов заданной матрицы размером 5х5.

**3. Ход работы**

На рисунках 1 - 5 представлена структурная схема программы на С++.

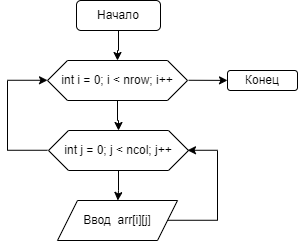


Рисунок 1 – Структурная-схема функции create\_arr().

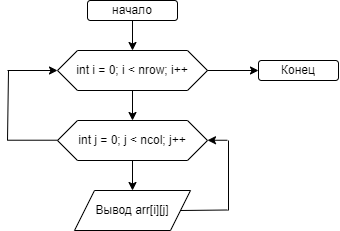


Рисунок 2 – Структурная схема функции output\_arr().

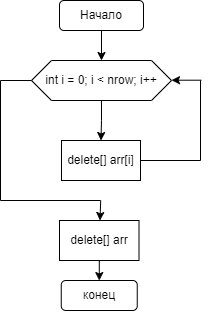


Рисунок 3 – Структурная схема функции delete\_arr().

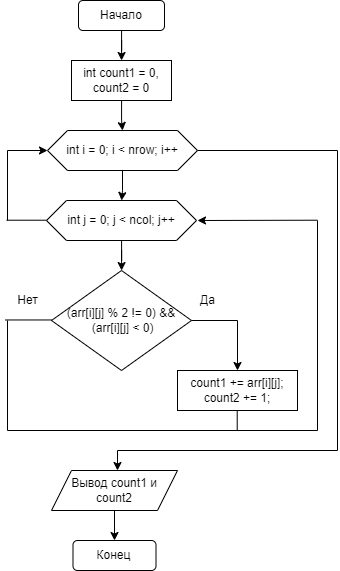


Рисунок 4 – Структурная схема функции num\_elem\_arr().

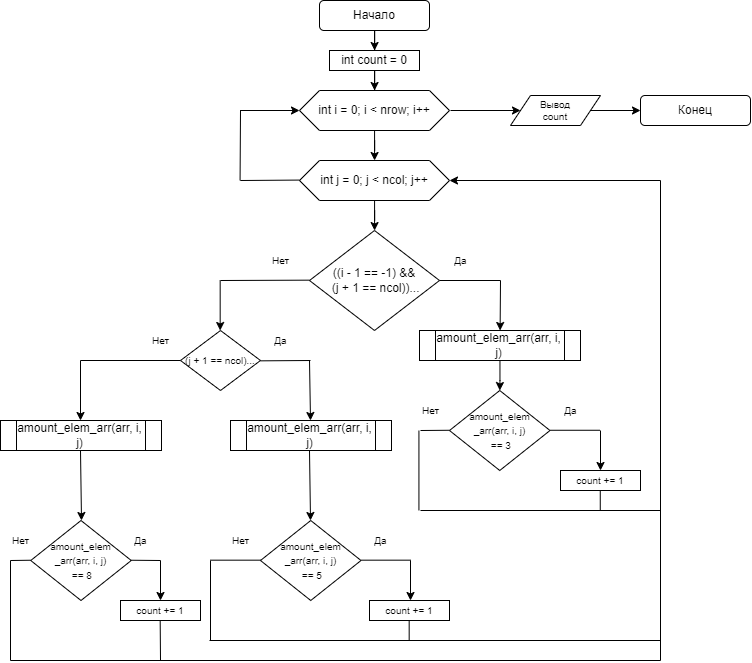
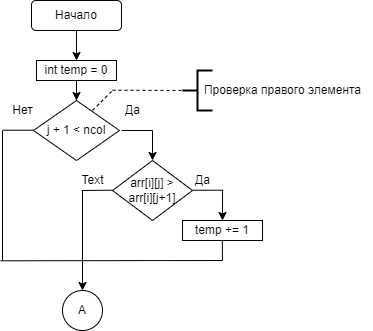
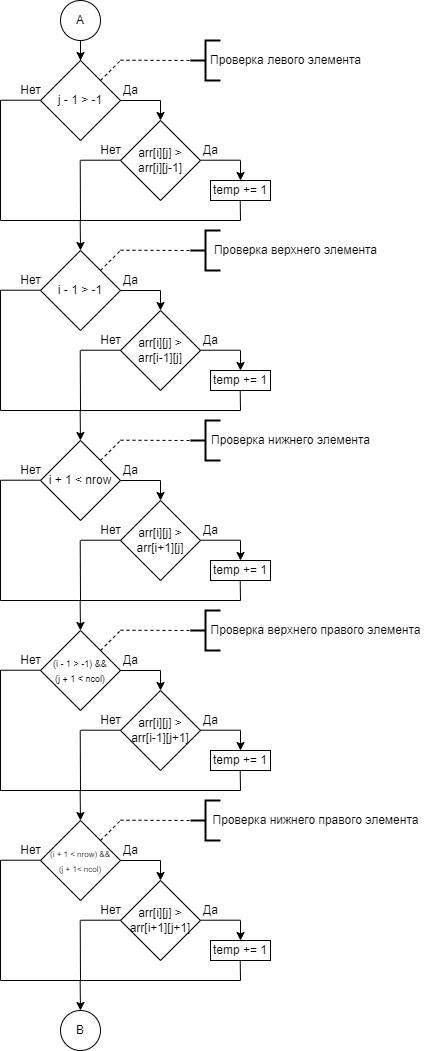


Рисунок 5 – Структурная схема функции local\_max\_arr().





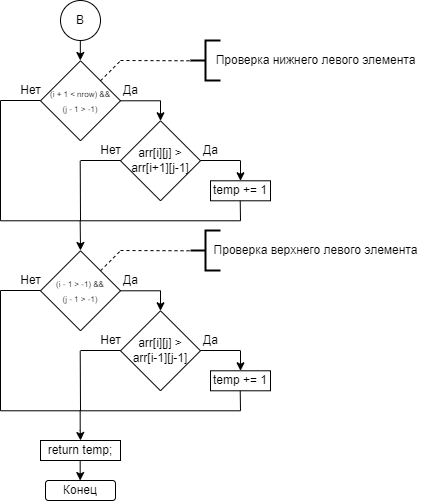
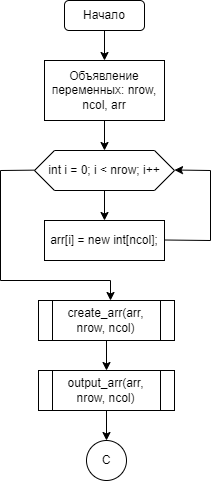


Рисунок 6 – Структурная схема функции amount\_elem\_arr().



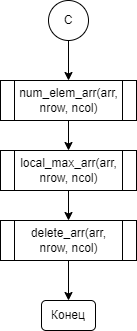


Рисунок 6 – Структурная схема работы программы.

На основе структурной-схемы была написана программа на языке C++.

В листинге 1 представлен код написанной программы.

Листинг 1 – Исходный код программы на C++.

#include <iostream>

#include <windows.h>

using namespace std;

const unsigned int nrow = 5, ncol = 5;

void create\_arr(int\*\* arr, int nrow, int ncol);

void output\_arr(int\*\* arr, int nrow, int ncol);

void delete\_arr(int\*\* arr, int nrow, int ncol);

void num\_elem\_arr(int\*\* arr, int nrow, int ncol);

void local\_max\_arr(int\*\* arr, int nrow, int ncol);

int amount\_elem\_arr(int\*\* arr, int i, int j);

int main() {

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

int\*\* arr = new int\*[nrow];

for (int i = 0; i < nrow; i++) {

arr[i] = new int[ncol];

}

cout << "### Поиск локальных максимумов матрицы 5х5 ###" << endl;

cout << "1) Найти количество и сумму нечётных элементов матрицы." << endl;

cout << "2) Найти количество локальных максимумов матрицы." << endl;

system("pause");

cout << "### Ввод элементов матрицы ###" << endl;

system("pause");

create\_arr(arr, nrow, ncol);

cout << "### Итоговая матрица: " << endl;

output\_arr(arr, nrow, ncol);

num\_elem\_arr(arr, nrow, ncol);

system("pause");

local\_max\_arr(arr, nrow, ncol);

delete\_arr(arr, nrow, ncol);

cout << "### Конец ###" << endl;

system("pause");

return 0;

}

void create\_arr(int\*\* arr, int nrow, int ncol) {

for (int i = 0; i < nrow; i++){

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

// arr[i][j] = rand() % 20;

cout << "Input arr[ " << i << " ][ " << j << " ]: " ; cin >> arr[i][j];

}

}

cout << endl;

}

void output\_arr(int\*\* arr, int nrow, int ncol) {

for (int i = 0; i < nrow; i++) {

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

cout << arr[i][j] << "\t";

}

cout << endl;

}

cout << endl;

}

void delete\_arr(int\*\* arr, int nrow, int ncol) {

for (int i = 0; i < nrow; i++) {

delete[] arr[i];

}

delete[] arr;

}

void num\_elem\_arr(int\*\* arr, int nrow, int ncol) {

int count1 = 0, count2 = 0;

for (int i = 0; i < nrow; i++) {

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

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

count1 += arr[i][j];

count2 += 1;

}

}

}

cout << "### Количество нечётных отрицательных элементов матрицы: " << count2 << endl;

cout << "### Сумма нечётных отрицательных элементов матрицы: " << count1 << endl;

cout << endl;

}

void local\_max\_arr(int\*\* arr, int nrow, int ncol) {

int count = 0;

for (int i = 0; i < nrow; i++) {

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

if (((i - 1 == -1) && (j + 1 == ncol))

|| ((i + 1 == nrow) && (j + 1 == ncol))

|| ((i + 1 == nrow) && (j - 1 == -1))

|| ((i - 1 == -1) && (j - 1 == -1))) { // Угловые элементы матрицы

if (amount\_elem\_arr(arr, i, j) == 3) count += 1;

} else if ((j + 1 == ncol) || (j - 1 == -1)

|| (i - 1 == -1) || (i + 1 == nrow)) { // Боковые элементы матрицы

if (amount\_elem\_arr(arr, i, j) == 5) count += 1;

} else {

if (amount\_elem\_arr(arr, i, j) == 8) count += 1;

}

}

}

cout << "### Количество локальных максимумов матрицы: " << count << endl;

}

int amount\_elem\_arr(int\*\* arr, int i, int j) {

int temp = 0;

if (j + 1 < ncol) { // Правый элемент

if (arr[i][j] > arr[i][j+1]) {

temp += 1;

}

}

if (j - 1 > -1) { // Левый элемент

if (arr[i][j] > arr[i][j-1]) {

temp += 1;

}

}

if (i - 1 > -1) { // Верхний элемент

if (arr[i][j] > arr[i-1][j]) {

temp += 1;

}

}

if (i + 1 < nrow) { // Нижний элемент

if (arr[i][j] > arr[i+1][j]) {

temp += 1;

}

}

if ((i - 1 > -1) && (j + 1 < ncol)) { // Верхний правый элемент

if (arr[i][j] > arr[i-1][j+1]) {

temp += 1;

}

}

if ((i + 1 < nrow) && (j + 1 < ncol)) { // Нижний правый элемент

if (arr[i][j] > arr[i+1][j+1]) {

temp += 1;

}

}

if ((i + 1 < nrow) && (j - 1 > -1)) { // Нижний левый элемент

if (arr[i][j] > arr[i+1][j-1]) {

temp += 1;

}

}

if ((i - 1 > -1) && (j - 1 > -1)) { // Верхний левый элемент

if (arr[i][j] > arr[i-1][j-1]) {

temp += 1;

}

}

return temp;

}

Далее было проведено тестирование программы. Были взяты три различных набора данных для тестирования программы(Рисунки 7-9).

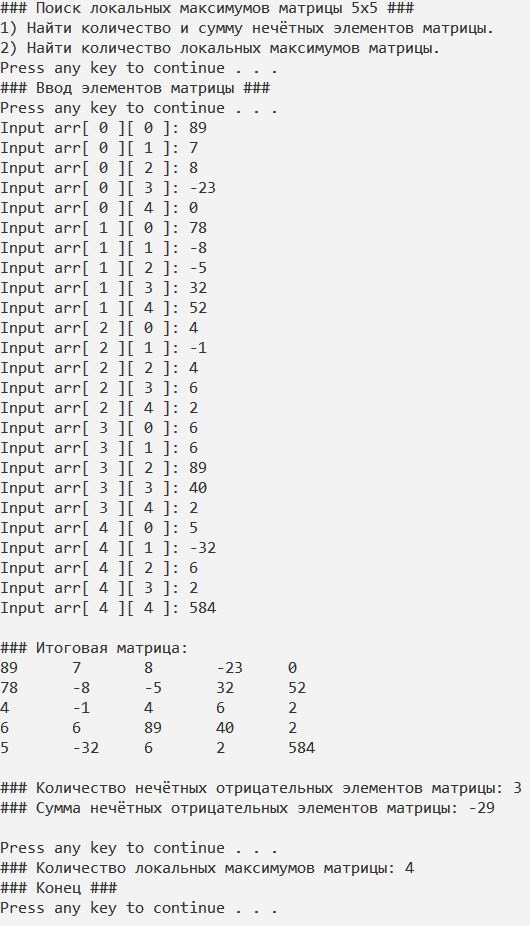


Рисунок 7 – Тестирование программы на C++

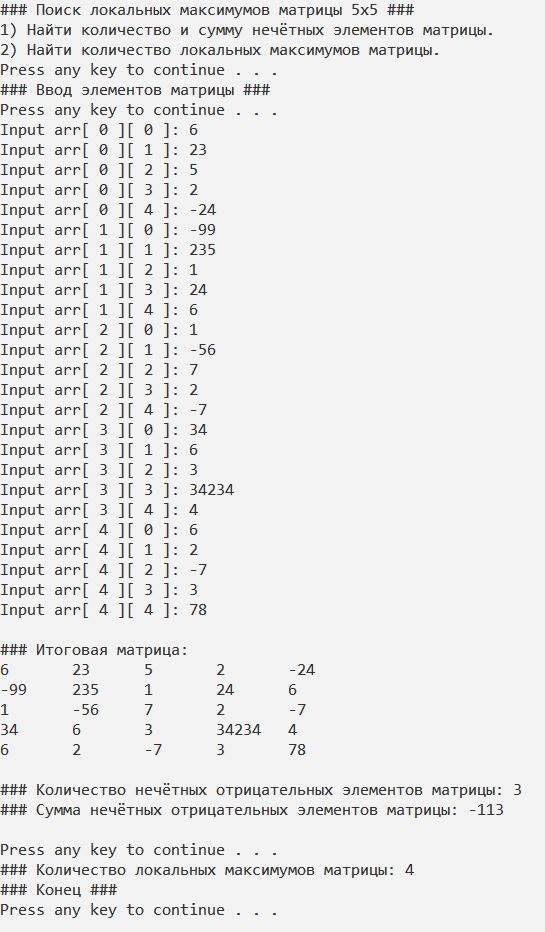


Рисунок 8 – Тестирование программы на C++

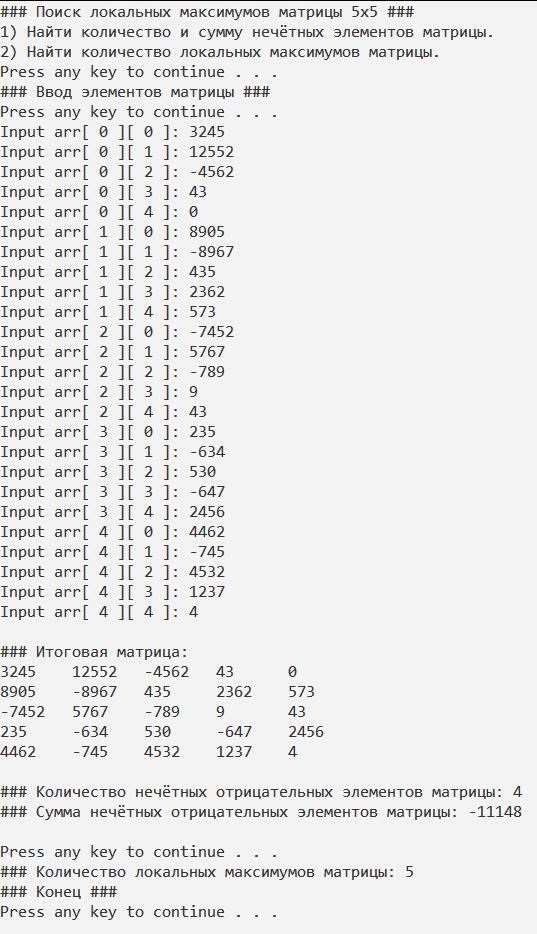


Рисунок 9 – Тестирование программы на C++

**Выводы**

Были изучены методы представления динамических двумерных массивов в C/C++, их заполнение, сортировка и удаление. Так же были рассмотрены способы передачи аргументов в функции.