*Лабораторная работа №9*

*студента группы ИТ-193*

*Кондратенко Виктора Витальевича*

*Выполнение: \_\_\_\_\_\_\_\_\_*

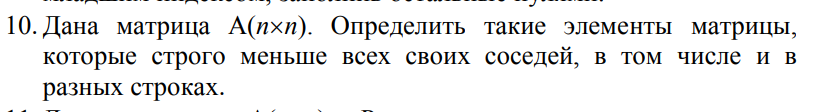
*Защита: \_\_\_\_\_\_\_\_\_\_*

Обработка динамических массивов

Цель работы: ознакомиться с организацией многомерных динамических массивов в языке С/С++.

***Содержание работы***

1. Ознакомиться с теоретическим материалом.
2. Выбрать алгоритм, составить его блок-схему и программу, выполняющую создание и обработку двумерного динамического массива, в соответствии со своим вариантом задания (см. рис. Рис. 1. ). Во всех вариантах предполагается, что размерность массива задается на этапе выполнения пользователем. Элементы матрицы вводятся с клавиатуры. На экран выводится исходная матрица и результаты работы программы.

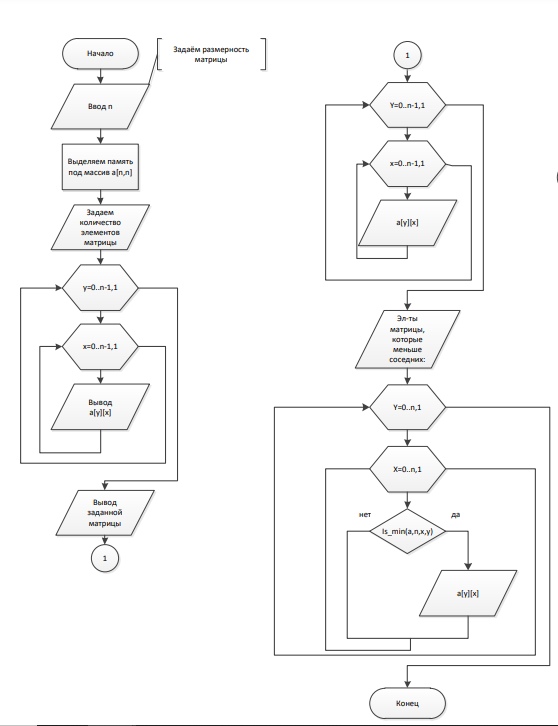


Задание для варианта 10

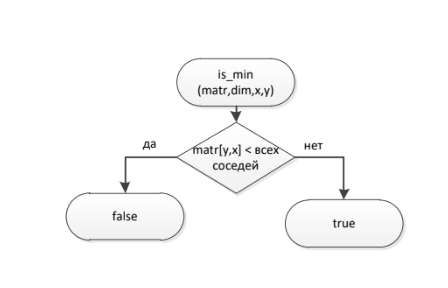
***Ход работы***

Вариант 10

1. Блок-схема (см. рисунки Рис. 2. -Рис. 3. ):



Основная блок-схема



Функция проверки выхода за границы массива

1. Программа

#include <stdio.h>

#include <locale.h>

#include <conio.h>

bool is\_min(int\*\* matr, int dim, int x, int y) {

if (x > 0 && y > 0 && matr[y][x] >= matr[y - 1][x - 1]) return false;

if (y > 0 && matr[y][x] >= matr[y - 1][x]) return false;

if (x < (dim-1) && y > 0 && matr[y][x] >= matr[y - 1][x + 1]) return false;

if (x > 0 && matr[y][x] >= matr[y][x - 1]) return false;

if (x < (dim - 1) && matr[y][x] >= matr[y][x + 1]) return false;

if (x > 0 && y < (dim - 1) && matr[y][x] >= matr[y + 1][x - 1]) return false;

if (y < (dim - 1) && matr[y][x] >= matr[y + 1][x]) return false;

if (x < (dim - 1) && y < (dim - 1) && matr[y][x] >= matr[y + 1][x + 1]) return false;

return true;

};

int main() {

setlocale(LC\_CTYPE, "");

int n;

printf("Задайте размерность матрицы a[n][n], n=\n");

scanf\_s("%d", &n);

int\*\* a, x, y;

a = new int\* [n];

printf("Задайте %d элементов матрицы (в матрице они будут построчно)\n", n \* n);

for (y = 0; y < n; y++) {

a[y] = new int[n];

for (x = 0; x < n; x++) {

scanf\_s("%d", &a[y][x]);

}

}

printf("Задана матрица a[%d][%d]:\n", n, n);

for (y = 0; y < n; y++) {

for (x = 0; x < n; x++) {

printf("%d\t", a[y][x]);

}

printf("\n");

}

printf("Элеметы матрицы, которые меньше соседних:\n");

for (y = 0; y < n; y++) {

for (x = 0; x < n; x++) {

if (is\_min(a, n, x, y)) {

printf("a[%d][%d]=%d\t", y, x, a[y][x]);

}

}

printf("\n");

}

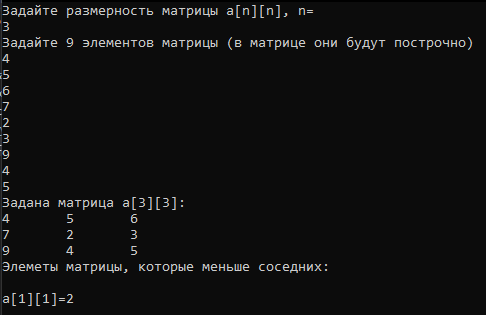
delete[]a;

\_getch();

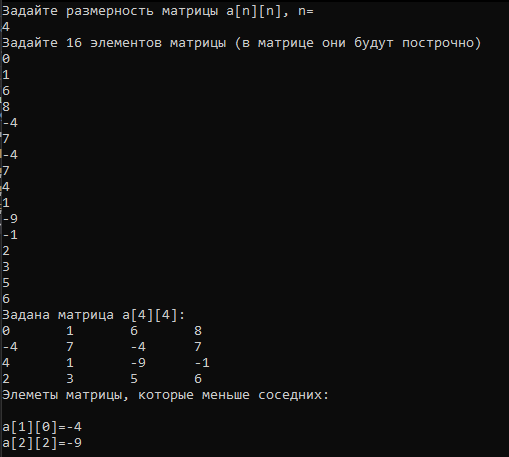
return 0;

}

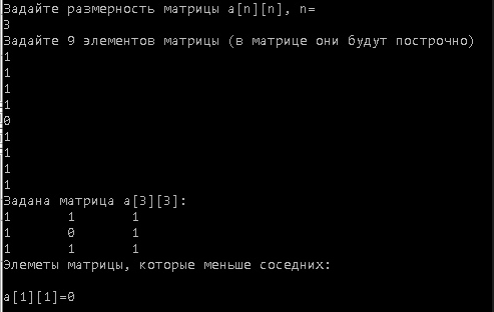
1. Тестирование программы:



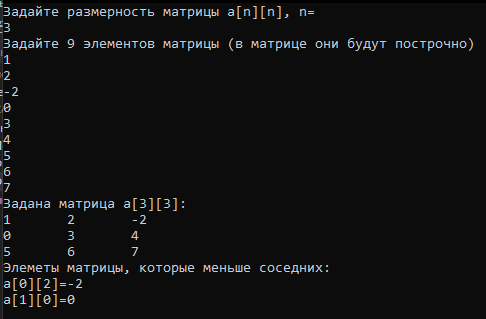
Тест №1



Тест №2



Тест №3



Тест №4

Вывод: в ходе выполнения данной работы удалось научиться работать с оператором new, позволяющим размещать массивы и оператором delete, позволяющим освобождать динамически размещенный массив.