МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №8

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Статические двумерные массивы»

Работу выполнил

студент гр. 4941 А. Комолкин

Санкт-Петербург

2021

***Цель лабораторной работы:*** *изучение структурной организации массивов и способов доступа к их элементам; совершенствование навыков процедурного программирования на языке C/C++ при решении задач обработки двумерных статических массивов (матриц).*

***Задание на программирование:*** *используя технологию процедурного программирования, разработать программу обработки двумерных статических массивов (матриц) в соответствии с индивидуальным заданием.*

***Порядок выполнения работы:***

1. Получить у преподавателя индивидуальное задание и выполнить постановку задачи: сформулировать условие, определить входные и выходные данные, их ограничения.

2. Разработать математическую модель: описать с помощью формул и рисунков структуру массивов и процесс их преобразования.

3. Построить схемы алгоритмов основных функций и задачи в целом.

4. Составить программу на языке *C/C++*.

5. Входные данные на этапах тестирования и демонстрации работы преподавателю должны задаваться либо с использованием специально подобранных арифметических формул, либо вводиться с клавиатуры по запросу. **Датчики псевдослучайных чисел использовать запрещается**.

Все значения элементов исходной матрицы **должны быть разными**.

Выходные данные должны выводиться на экран с пояснениями.

6. Проверить и продемонстрировать преподавателю работу программы на полном наборе тестов, в том числе с ошибочными входными данными. Входные и выходные массивы должны выводиться в одном и том же формате.

7. Использовать стандартные потоковые объекты ввода/вывода ***cin*** и ***cout***.

8. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, схемы алгоритмов, текст программы, контрольные примеры (скриншоты).

**Задание 31**

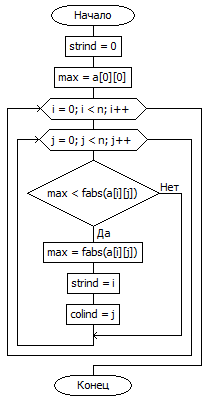
В заданной квадратной матрице размера (2n+1)×(2n+1) найти элементы с наибольшим по абсолютной величине и с наименьшим по абсолютной величине значениями. Если хотя бы один из них находится в области правее главной и побочной диагоналей найти сумму значений элементов, лежащих ниже главной диагонали. В противном случае поменять найденные значения местами.

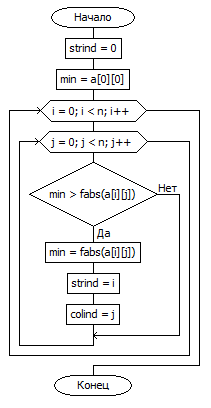
**Математическая модель**

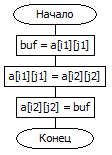
Необходимо реализовать поиск строчки с минимальным и максимальным значением путем перебора всех элементов матрицы.

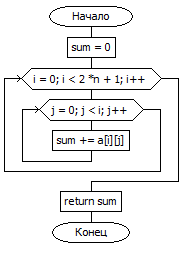
Условие вхождение элемента в области правее главной и побочной диагоналей: j> I and j > 2\*n - i. Задача решена.

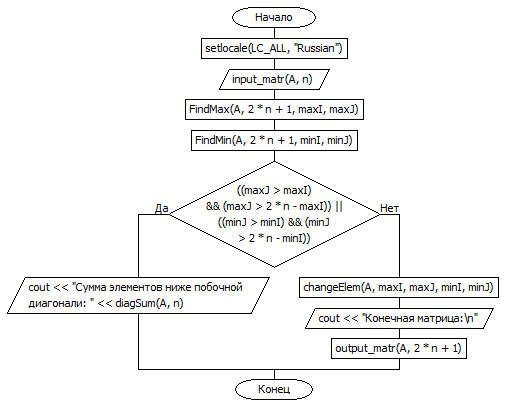
**Блок-схема алгоритма:**

****

****

****

****

****

**Текст программы**

#include <iostream>

using namespace std;

const int RAZ = 13; //максимальный размер квадратной матрицы

typedef int telem; //определение типа значений элементов массива

typedef telem tmatr[RAZ][RAZ]; //определение типа массива

typedef telem tarr[RAZ]; //определение типа одномерного массива

void input\_matr(tmatr a, int& n);

void output\_matr(tmatr a, int n);

void FindMax(tmatr a, int n, int& strind, int& colind);

void FindMin(tmatr a, int n, int& strind, int& colind);

telem diagSum(tmatr a, int n);

void changeElem(tmatr& a, int i1, int j1, int i2, int j2);

int main()

{

setlocale(LC\_ALL, "Russian");

int n, maxI, maxJ, minI, minJ;

tmatr A;

input\_matr(A, n);

FindMax(A, 2\*n+1, maxI, maxJ);

FindMin(A, 2 \* n + 1, minI, minJ);

if (((maxJ > maxI) && (maxJ > 2 \* n - maxI)) || ((minJ > minI) && (minJ > 2 \* n - minI)))

{

cout << "Сумма элементов ниже побочной диагонали: " << diagSum(A, n);

}

else

{

changeElem(A, maxI, maxJ, minI, minJ);

cout << "Конечная матрица:\n";

output\_matr(A, 2\*n+1);

}

}

//main\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//input\_matr\_beg\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Ввод исходных данных

void input\_matr(tmatr a, int& n)

{

cout << "Введите половину размера квадратной матрицы < " << (RAZ - 1) / 2 << " ";

cin >> n;

cout << "Введите построчно через пробел элементы" << endl;

cout << "матрицы размера " << 2\*n+1 << "x" << 2 \* n + 1 << endl;

cout << "После ввода строки нажимайте <Enter>" << endl;

for (int i = 0; i < 2 \* n + 1; i++)

for (int j = 0; j < 2 \* n + 1; j++)

cin >> a[i][j];

}

//input\_matr\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//output\_matr\_beg\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Вывод матрицы

void output\_matr(tmatr a, int n)

{

for (int i = 0; i < n; i++)

{

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

{

cout.width(3);

cout << a[i][j];

}

cout << endl;

}

return;

}

//output\_matr\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void FindMax(tmatr a, int n, int& strind, int & colind)

{

strind = 0;

telem max = a[0][0];

for (int i = 0; i < n; i++)

{

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

{

if (max < fabs(a[i][j]))

{

max = fabs(a[i][j]);

strind = i;

colind = j;

}

}

}

}

void FindMin(tmatr a, int n, int& strind, int& colind)

{

strind = 0;

telem min = a[0][0];

for (int i = 0; i < n; i++)

{

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

{

if (min > fabs(a[i][j]))

{

min = fabs(a[i][j]);

strind = i;

colind = j;

}

}

}

}

telem diagSum(tmatr a, int n)

{

telem sum = 0;

for (int i = 0; i < 2 \* n + 1; i++)

{

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

{

sum += a[i][j];

}

}

return sum;

}

void changeElem(tmatr& a, int i1, int j1, int i2, int j2)

{

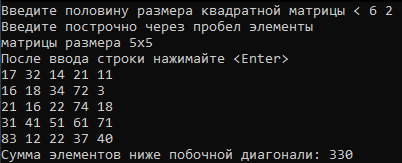
telem buf = a[i1][j1];

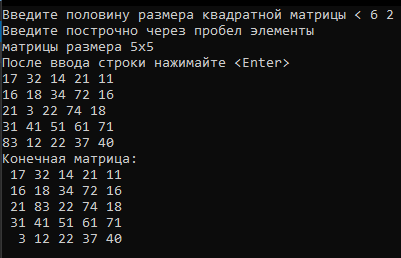
a[i1][j1] = a[i2][j2];

a[i2][j2] = buf;

}

**Пример работы программы**

****

****

**Вывод:** используя технологию процедурного программирования, разработал программу обработки двумерных статических массивов (матриц) в соответствии с индивидуальным заданием.