Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н.Ельцина»

ИРИТ-РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ N 2**

по дисциплине «Современные языки программирования»

Студент группы РИВ-210938у: Вахрушев Н.А.

Преподаватель: Архипов Н.А.

Екатеринбург 2023

Цель

Необходимо приобрести навыки работы с матричным представлением данных.

Ход работы

Ниже представлены блок схемы разработанных алгоритмов:

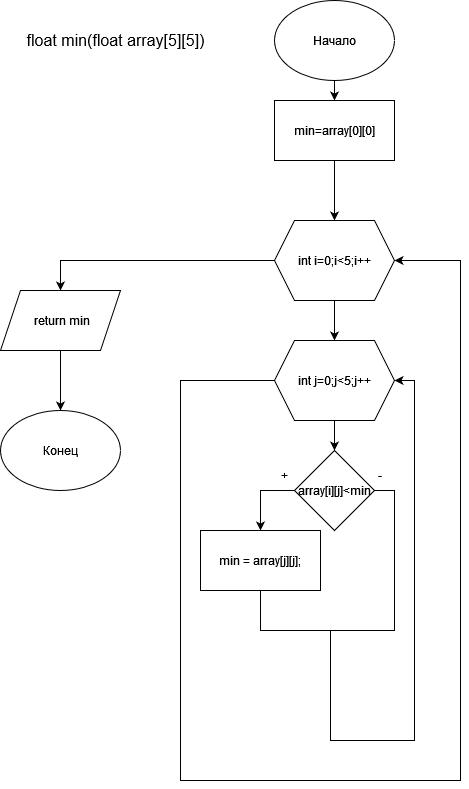


Рисунок 1 – Блок схема функций min

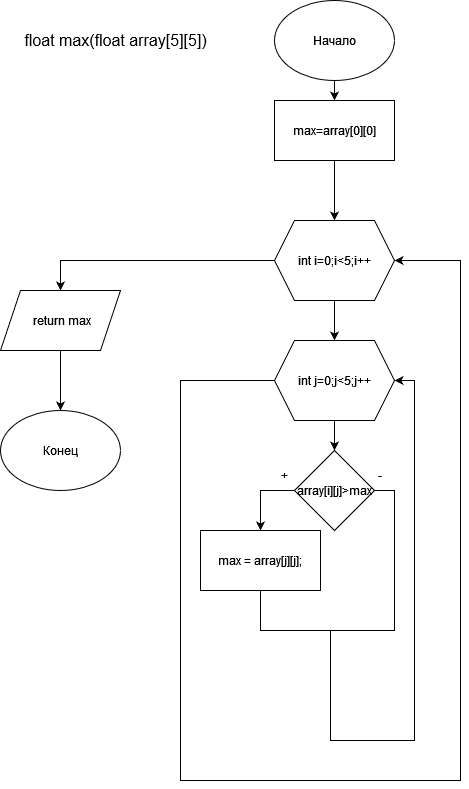


Рисунок 2 – Блок схема функций max

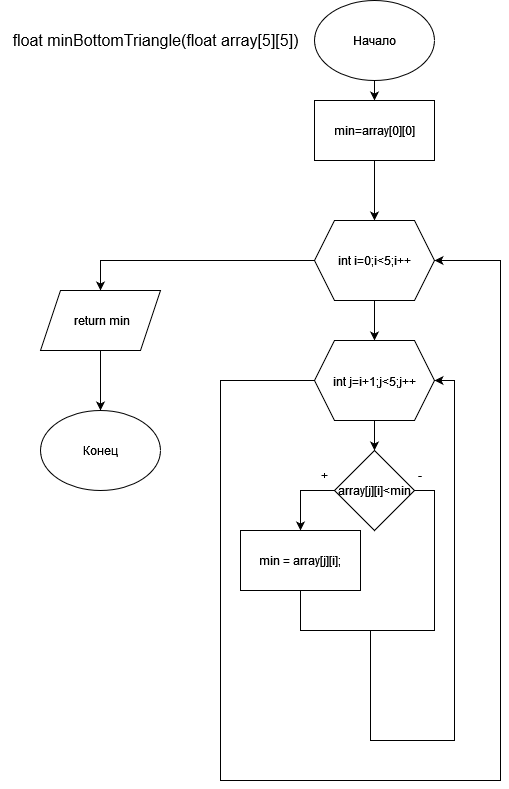


Рисунок 3 – Блок схема функций minBottomTriangle

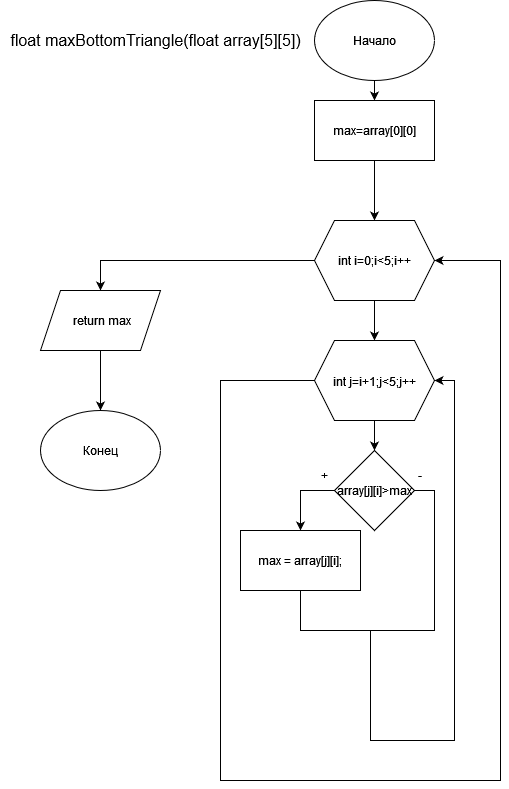


Рисунок 4 – Блок схема функций maxBottomTriangle

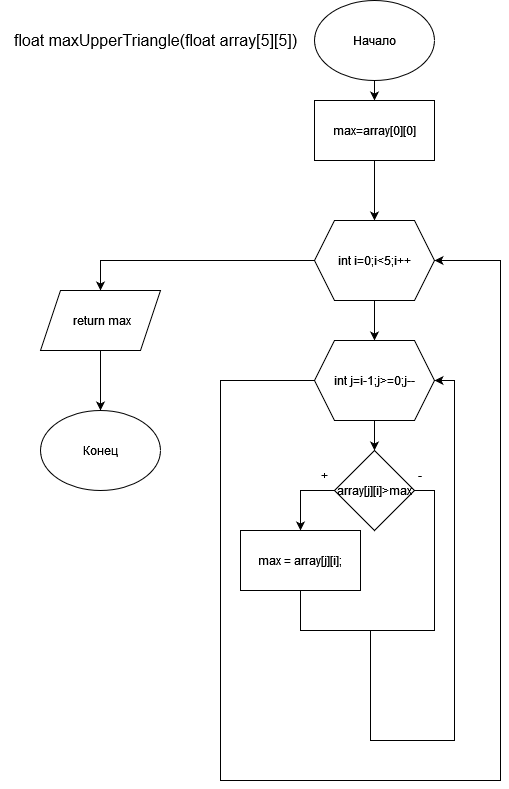


Рисунок 5 – Блок схема функций maxUpperTriangle

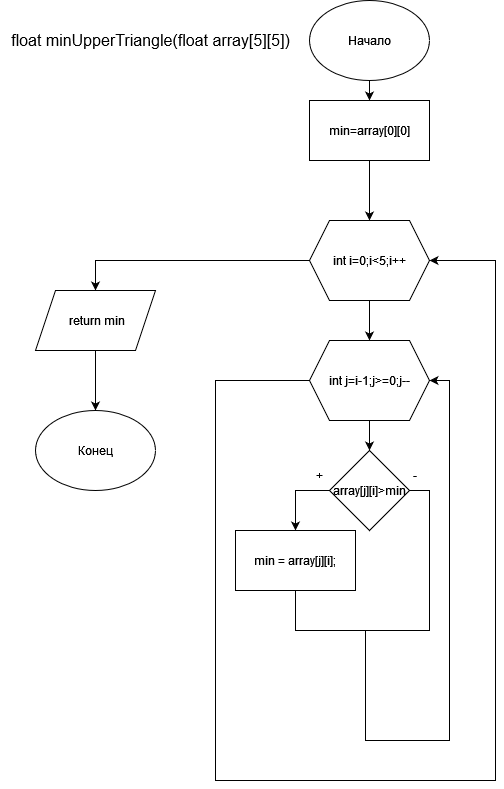


Рисунок 6 – Блок схема функций minUpperTriangle

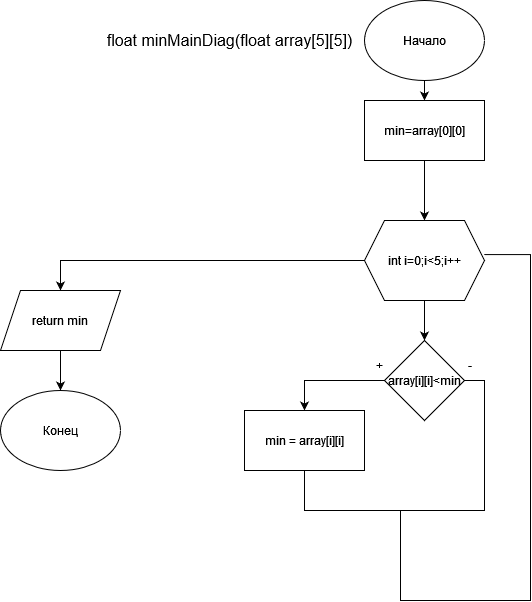


Рисунок 7 – Блок схема функций minMainDiag

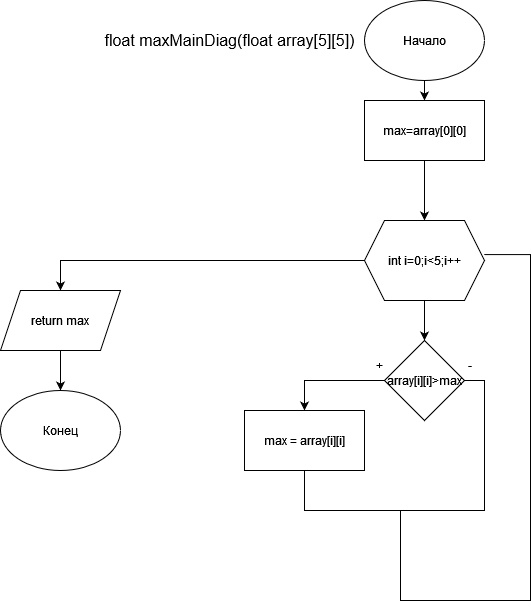


Рисунок 8 – Блок схема функций maxMainDiag

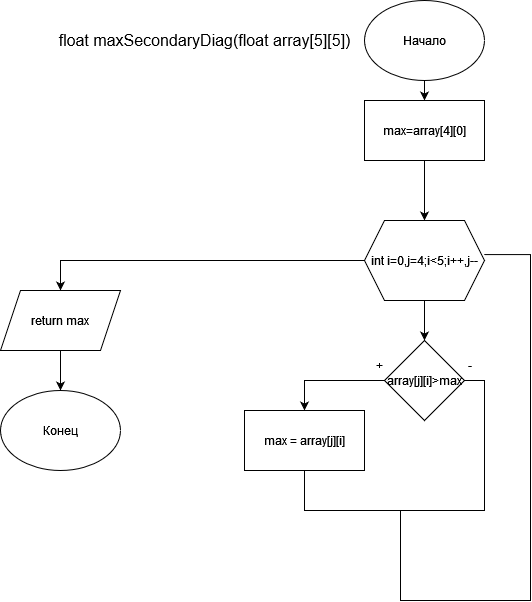


Рисунок 9 – Блок схема функций maxSecondaryDiag

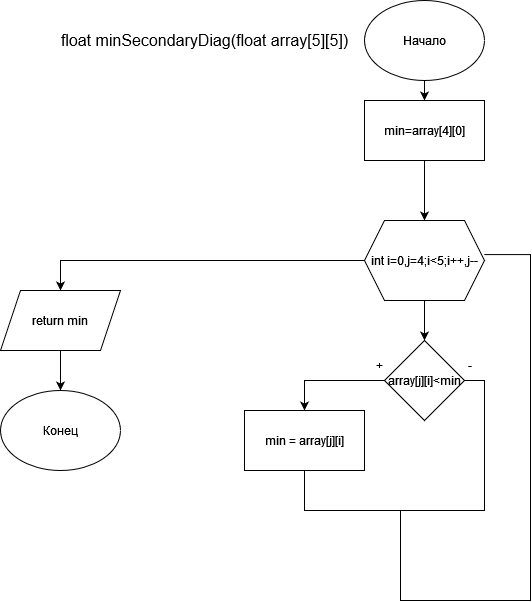


Рисунок 10 – Блок схема функций minSecondaryDiag

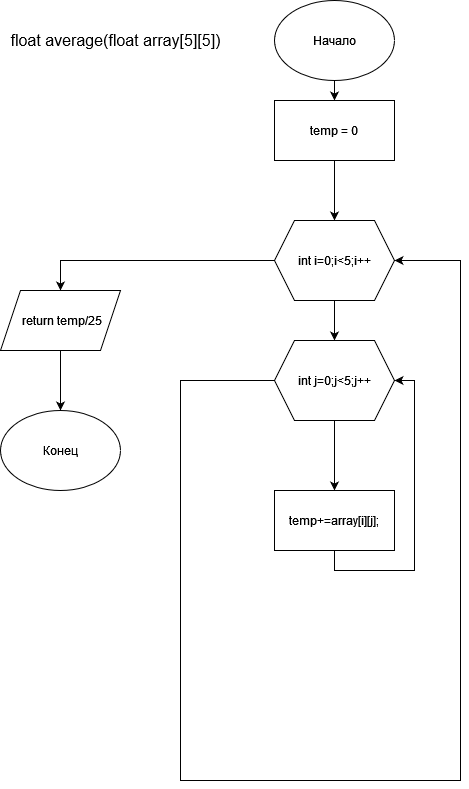


Рисунок 11 – Блок схема функций average

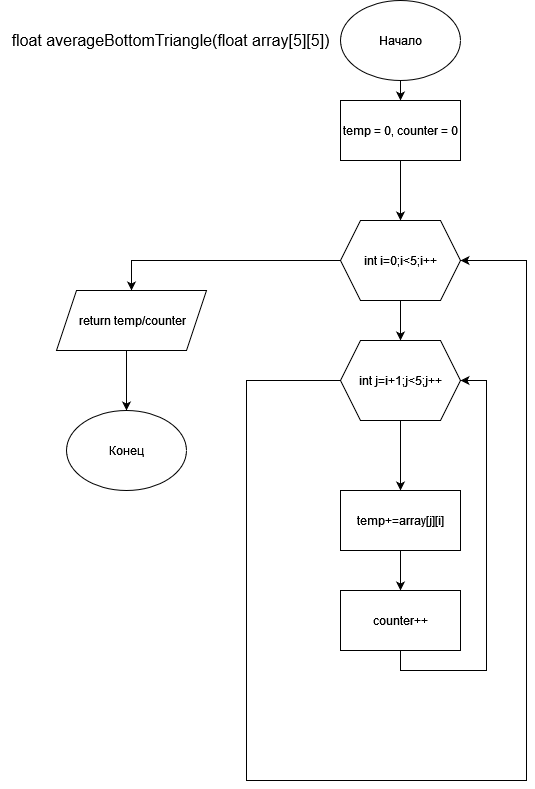


Рисунок 12 – Блок схема функций averageBottomTriangle

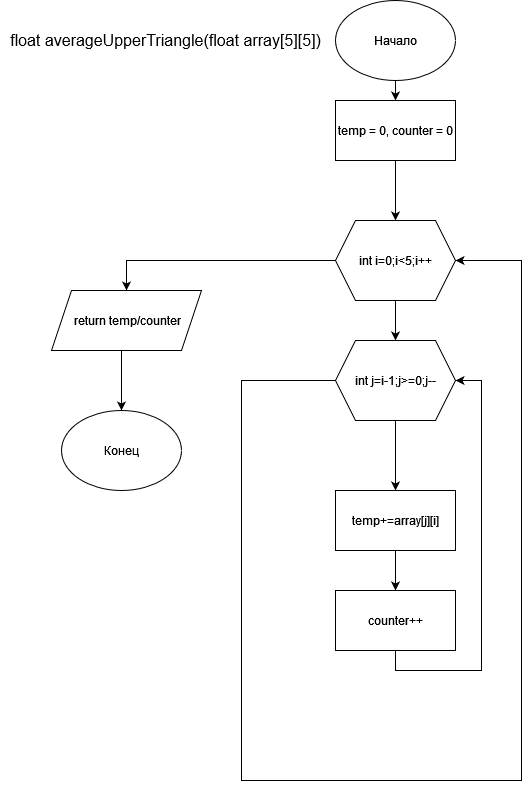


Рисунок 13 – Блок схема функций averageUpperTriangle

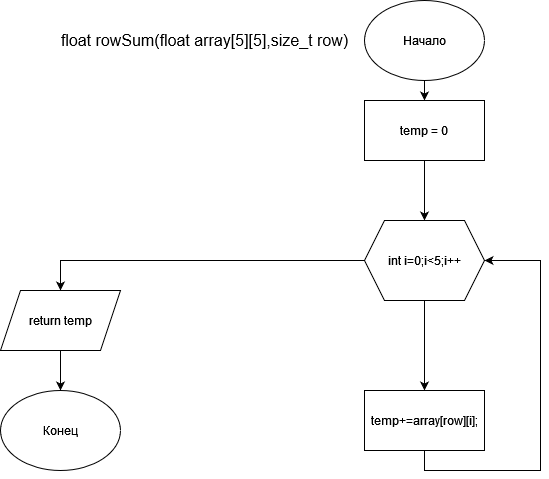


Рисунок 14 – Блок схема функцтй rowSum

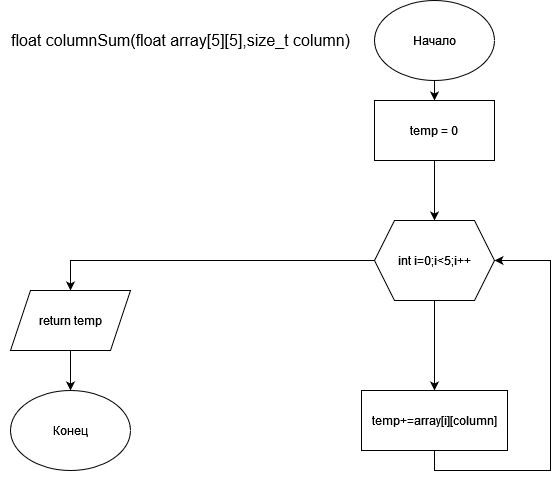


Рисунок 15 – Блок схема функций columnSum

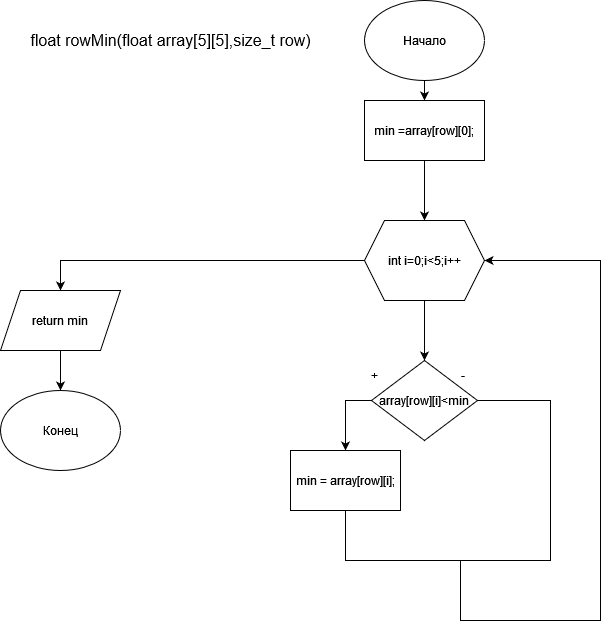


Рисунок 16 – Блок схема функций rowMin

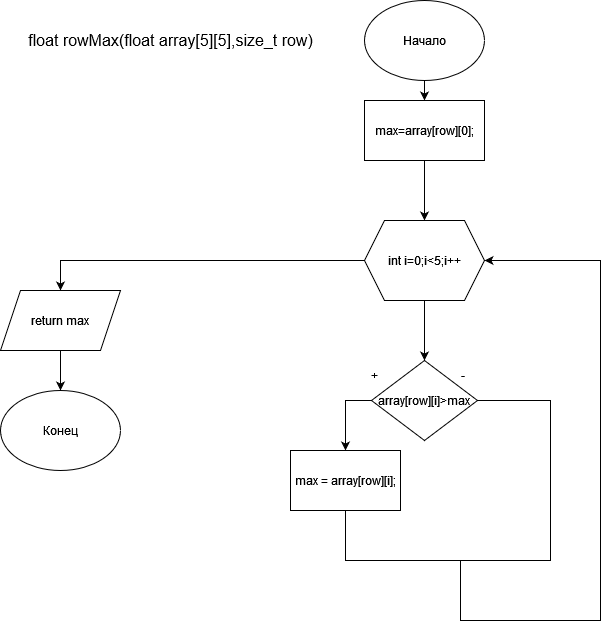


Рисунок 17 – Блок схема функций rowMax

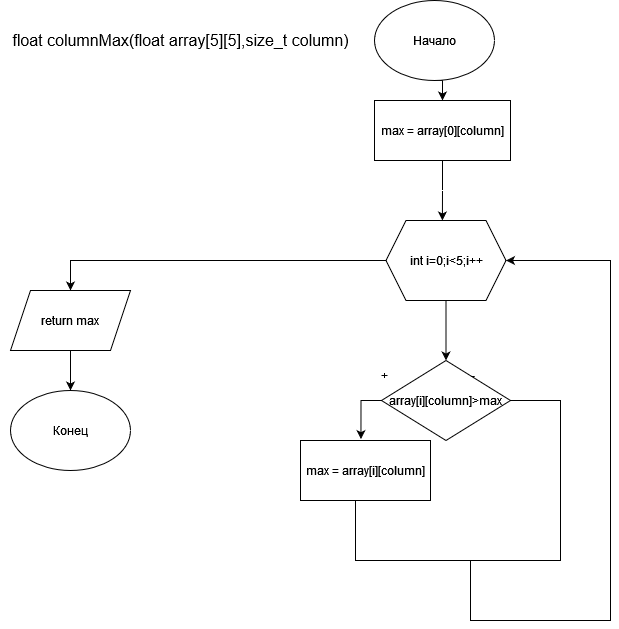


Рисунок 18 – Блок схема функций columnMax

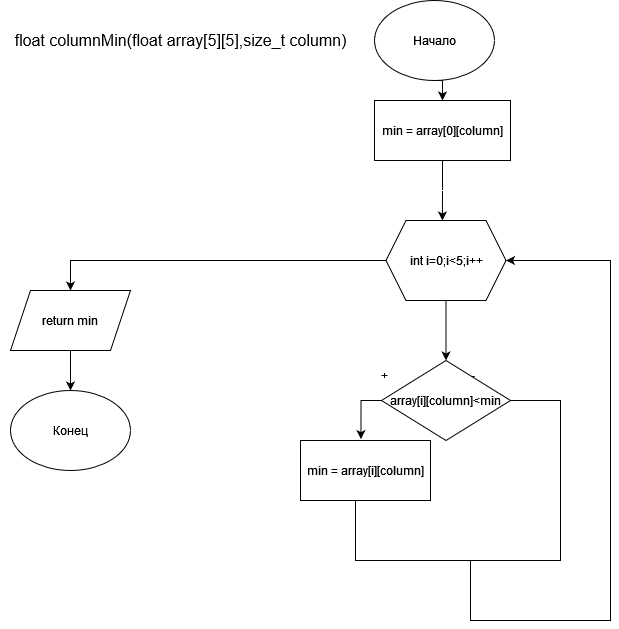


Рисунок 19 – Блок схема функций columnMin

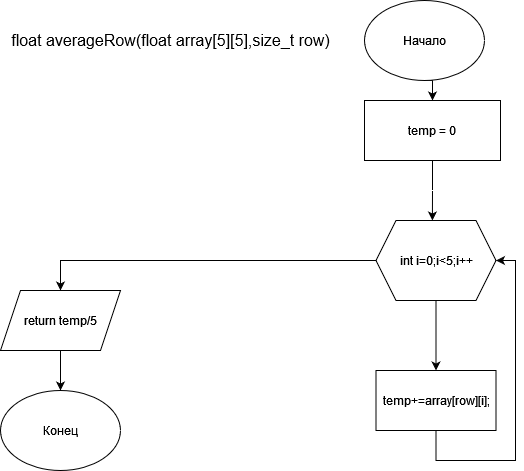


Рисунок 20 – Блок схема функций averageRow

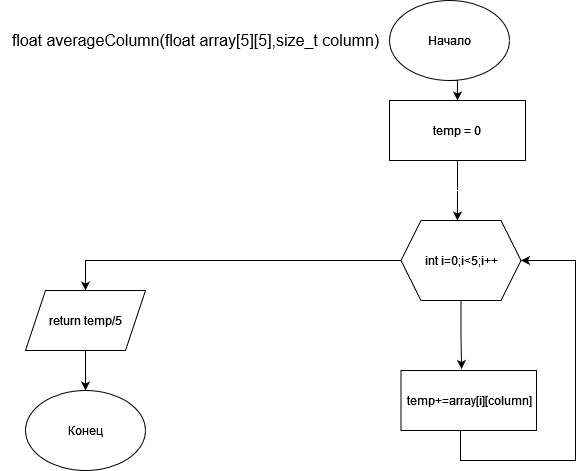


Рисунок 21 – Блок схема функций averageColumn

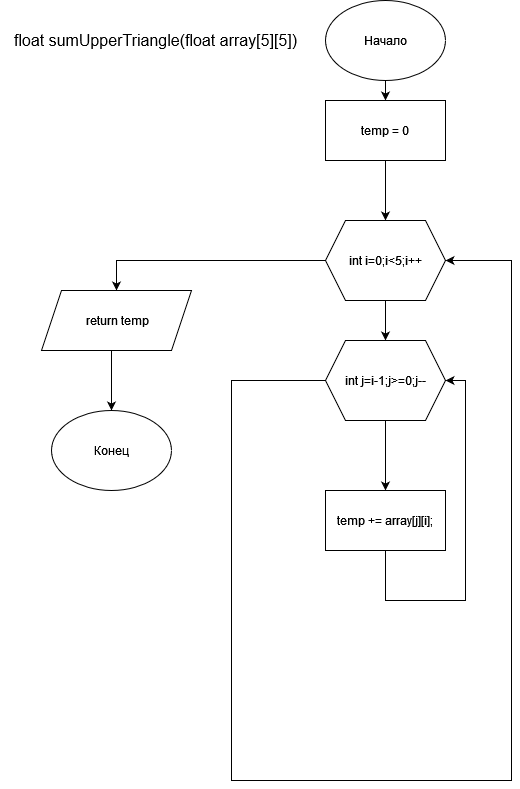


Рисунок 22 – Блок схема функций sumUpperTriangle

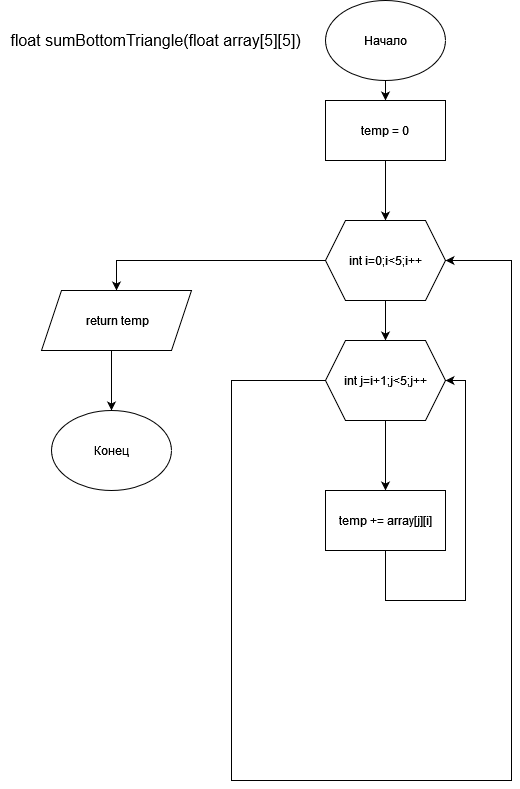


Рисунок 23 – Блок схема функций sumBottomTriangle

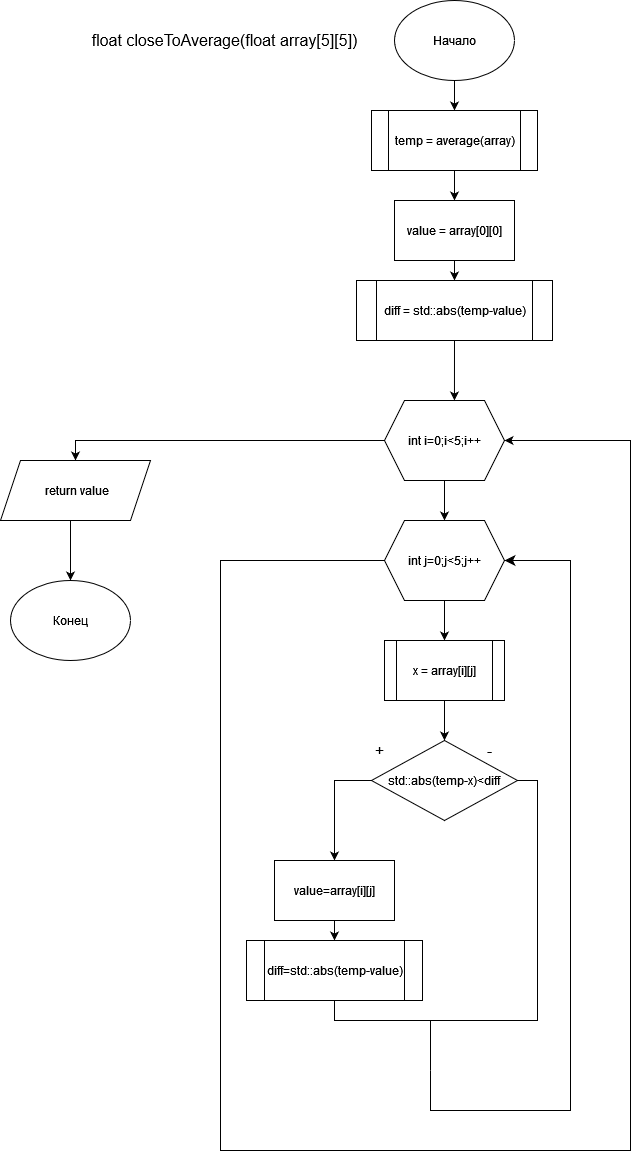


Рисунок 24 – Блок схема функций closeToAverage

Ниже представлен листинг программ:

#include <cmath>

//Нахождение минимального значения в матрице

float min(float array[5][5])

{

float min=array[0][0];

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

if(array[i][j]<min) min = array[i][j];

}

}

return min;

}

//Нахождение максимального значения в матрице

float max(float array[5][5])

{

float max=array[0][0];

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

if(array[i][j]>max) max = array[i][j];

}

}

return max;

}

//Нахождение минимального значения в нижнетреугольной части матрицы

float minBottomTriangle(float array[5][5])

{

float min=array[0][0];

for(int i=0;i<5;i++)

{

for(int j=i+1;j<5;j++)

{

if(array[j][i]<min) min = array[j][i];

}

}

return min;

}

//Нахождение максимального значения в нижнетреугольной части матрицы

float maxBottomTriangle(float array[5][5])

{

float max=array[0][0];

for(int i=0;i<5;i++)

{

for(int j=i+1;j<5;j++)

{

if(array[j][i]>max) max = array[j][i];

}

}

return max;

}

//Нахождение максимального значения в верхнетреугольной части матрицы

float maxUpperTriangle(float array[5][5])

{

float max=array[0][0];

for(int i=0;i<5;i++)

{

for(int j=i-1;j>=0;j--)

{

if(array[j][i]>max) max = array[j][i];

}

}

return max;

}

//Нахождение минимального значения в верхнетреугольной части матрицы

float minUpperTriangle(float array[5][5])

{

float min=array[0][0];

for(int i=0;i<5;i++)

{

for(int j=i-1;j>=0;j--)

{

if(array[j][i]<min) min = array[j][i];

}

}

return min;

}

//Нахождение минимального значения в главной диагонали

float minMainDiag(float array[5][5])

{

float min = array[0][0];

for(int i=0;i<5;i++)

{

if(array[i][i]<min) min=array[i][i];

}

return min;

}

//Нахождение максимального значения в главной диагонали

float maxMainDiag(float array[5][5])

{

float max = array[0][0];

for(int i=0;i<5;i++)

{

if(array[i][i]>max) max=array[i][i];

}

return max;

}

//Нахождение максимального значения в побочной диагонали

float maxSecondaryDiag(float array[5][5])

{

float max = array[4][0];

for(int i=0,j=4;i<5;i++,j--)

{

if(array[j][i]>max) max=array[j][i];

}

return max;

}

//Нахождение минимального значения в побочной диагонали

float minSecondaryDiag(float array[5][5])

{

float min = array[4][0];

for(int i=0,j=4;i<5;i++,j--)

{

if(array[j][i]<min) min=array[j][i];

}

return min;

}

//Нахождение среднеарефмитического значения

float average(float array[5][5])

{

float temp{0};

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

temp+=array[i][j];

}

}

return temp/25;

}

//Нахождение среднеарефмитического значения в нижнетреугольной части матрицы

float averageBottomTriangle(float array[5][5])

{

float temp{0};

size\_t counter{0};

for(int i=0;i<5;i++)

{

for(int j=i+1;j<5;j++)

{

temp+=array[j][i];

counter++;

}

}

return temp/counter;

}

//Нахождение среднеарефмитического значения в верхнетреугольной части матрицы

float averageUpperTriangle(float array[5][5])

{

float temp{0};

size\_t counter{0};

for(int i=0;i<5;i++)

{

for(int j=i-1;j>=0;j--)

{

temp+=array[j][i];

counter++;

}

}

return temp/counter;

}

//Нахождение суммы в конкретной строке

float rowSum(float array[5][5],size\_t row)

{

float temp {0};

for(int i=0;i<5;i++)

{

temp+=array[row][i];

}

return temp;

}

//Нахождение суммы в конкретном столбце

float columnSum(float array[5][5],size\_t column)

{

float temp {0};

for(int i=0;i<5;i++)

{

temp+=array[i][column];

}

return temp;

}

//Нахождение минимального значения в строке

float rowMin(float array[5][5],size\_t row)

{

float min =array[row][0];

for(int i=0;i<5;i++)

{

if(array[row][i]<min) min = array[row][i];

}

return min;

}

//Нахождение максимального значения в строке

float rowMax(float array[5][5],size\_t row)

{

float max=array[row][0];

for(int i=0;i<5;i++)

{

if(array[row][i]>max) max = array[row][i];

}

return max;

}

//Нахождение максимального значения в столбце

float columnMax(float array[5][5],size\_t column)

{

float max = array[0][column];

for(int i=0;i<5;i++)

{

if(array[i][column]>max) max = array[i][column];

}

return max;

}

//Нахождение минимального значения в столбце

float columnMin(float array[5][5],size\_t column)

{

float min = array[0][column];

for(int i=0;i<5;i++)

{

if(array[i][column]<min) min = array[i][column];

}

return min;

}

//Нахождение среднего арифметического значения в строке

float averageRow(float array[5][5],size\_t row)

{

float temp {0};

for(int i=0;i<5;i++)

{

temp+=array[row][i];

}

return temp/5;

}

//Нахождение среднего арифметического значения в столбце

float averageColumn(float array[5][5],size\_t column)

{

float temp {0};

for(int i=0;i<5;i++)

{

temp+=array[i][column];

}

return temp/5;

}

//Сумма чисел в верхнетреугольной матрице

float sumUpperTriangle(float array[5][5])

{

float temp{0};

for(int i=0;i<5;i++)

{

for(int j=i-1;j>=0;j--)

{

temp += array[j][i];

}

}

return temp;

}

//Сумма чисел в нижнетреугольной матрице

float sumBottomTriangle(float array[5][5])

{

float temp{0};

for(int i=0;i<5;i++)

{

for(int j=i+1;j<5;j++)

{

temp += array[j][i];

}

}

return temp;

}

//Нахождение числа, которое максимально близко к среднеарифметическому

float closeToAverage(float array[5][5])

{

float temp = average(array);

float value = array[0][0];

float diff = std::abs(temp-value);

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

float x = array[i][j];

if(std::abs(temp-x)<diff)

{

value=array[i][j];

diff=std::abs(temp-value);

}

}

}

return value;

}

Вывод

В результате работы были приобретены навыки работы с матричным представлением данных с помощью языка программирования C++.