**Домашняя работа № 20.02.2024г.**

## **По теме: «Многомерные массивы».**

### Задание 1:

Напишите программу, которая создает двухмерный массив и заполняет его по следующему принципу: пользователь вводит число (например, 3) первый элемент массива принимает значение этого числа, последующий элемент массива принимает значение этого числа умноженного на 2 (т.е. 6 для нашего примера), третий элемент массива предыдущий элемен,т умноженный на 2 (т.е. 6\*2=12 для нашего примера). Созданный массив вывести на экран.

### Решение:

#include <iostream>

#include <iomanip>

int main()

{

setlocale(LC\_ALL, "Russian");

const int size = 3;

int A[size][size];

int startNumber;

std::cout << "Введите первое число двумерного массива -> ";

std::cin >> startNumber;

std::cout << std::endl;

for (int i = 0; i < size; i++) // каждый последуюший элемент умножатся на 2

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

{

startNumber \*= 2;

A[i][j] = startNumber;

}

std::cout << "Массив А:" << std::endl;

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

{

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

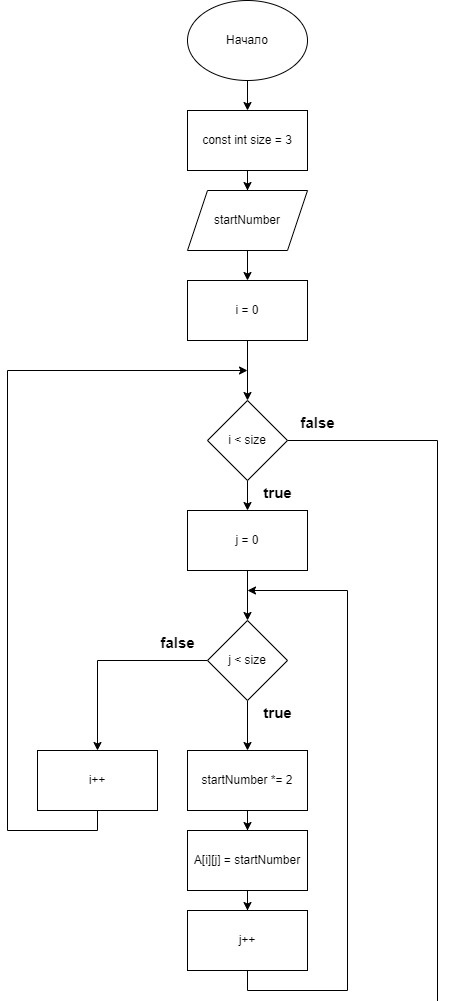
std::cout << std::setw(size+2) << A[i][j] << " ";

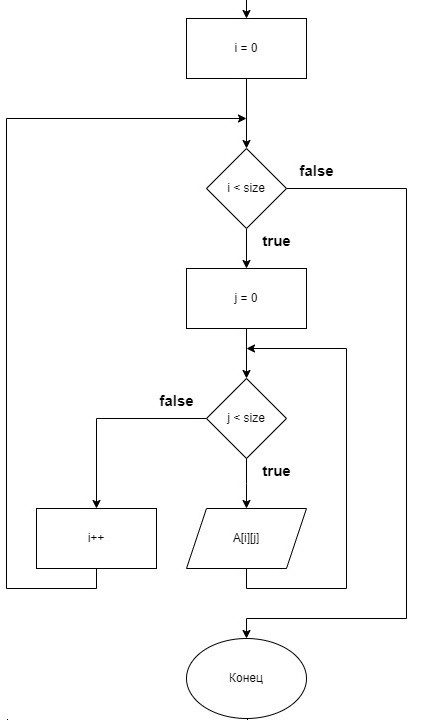
std::cout << std::endl;

}

}

### Блок-схема:





### Задание 2:

Напишите программу, которая создает двухмерный массив и заполняет его по следующему принципу: пользователь вводит число (например, 3) первый элемент массива принимает значение этого числа, последующий элемент массива принимает значение этого числа+ 1 (т.е. 4 для нашего примера), третий элемент массива предыдущий элемент + 1 (т.е. 5 для нашего примера). Созданный массив вывести на экран.

### Решение:

#include <iostream>

#include <iomanip>

int main()

{

setlocale(LC\_ALL, "Russian");

const int size = 5;

int A[size][size];

int startNumber;

std::cout << "Введите первый элемент массива -> ";

std::cin >> startNumber;

std::cout << std::endl;

std::cout << "\tМассива А:" << std::endl;

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

{

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

{

startNumber += 1;

A[i][j] = startNumber;

std::cout << std::setw(3) << A[i][j] << " ";

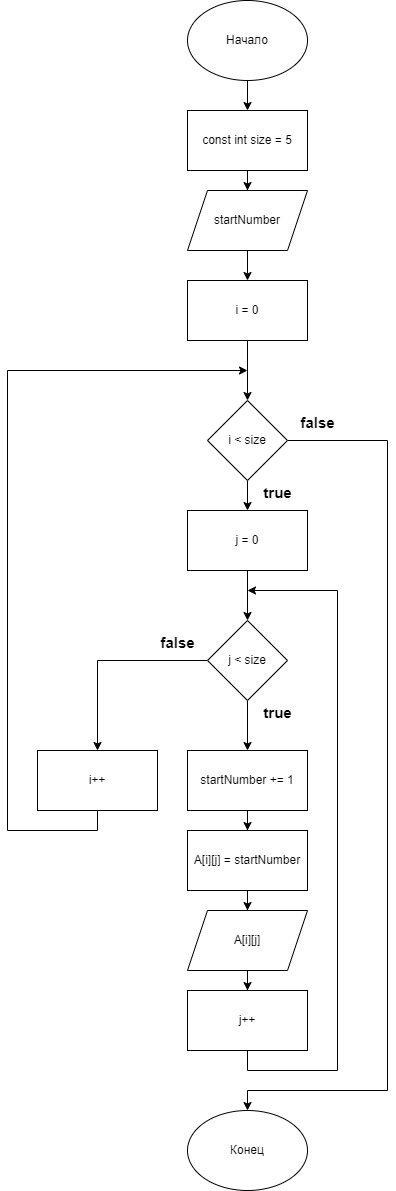
}

std::cout << std::endl;

}

}

### Блок-схема:



### Задание 4:

В двумерном массиве целых числе посчитать: Сумму всех элементов массива; Среднее арифметическое всех элементов массива; Минимальный элемент; Максимальный элемент

Решение:

#include <iostream>

#include <iomanip>

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(nullptr));

const int size = 5;

int A[size][size];

int sumElements = 0;

double averageNumber = 0.0;

int minElement;

int maxElement;

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

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

{

A[i][j] = rand() % 101; // [0; 101)

}

std::cout << "\tМассив А:" << std::endl;

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

{

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

std::cout << std::setw(3) << A[i][j] << " ";

std::cout << std::endl;

}

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

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

sumElements += A[i][j];

std::cout << "Сумму всех элементов массива: " << sumElements << std::endl;

averageNumber = (double)sumElements / ((double)size \* (double)size);

std::cout << "Среднее арифметическое всех элементов массива: " << averageNumber << std::endl;

minElement = A[0][0]; // пусть первый элемент массива будет минимальным значением

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

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

if (minElement > A[i][j]) // если последующий элемент массива будет меньше текущего, то minElement примет новое значение

minElement = A[i][j];

std::cout << "Минимальный элемент: " << minElement << std::endl;

maxElement = A[0][0]; // пусть первый элемент массива будет максимальным значением

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

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

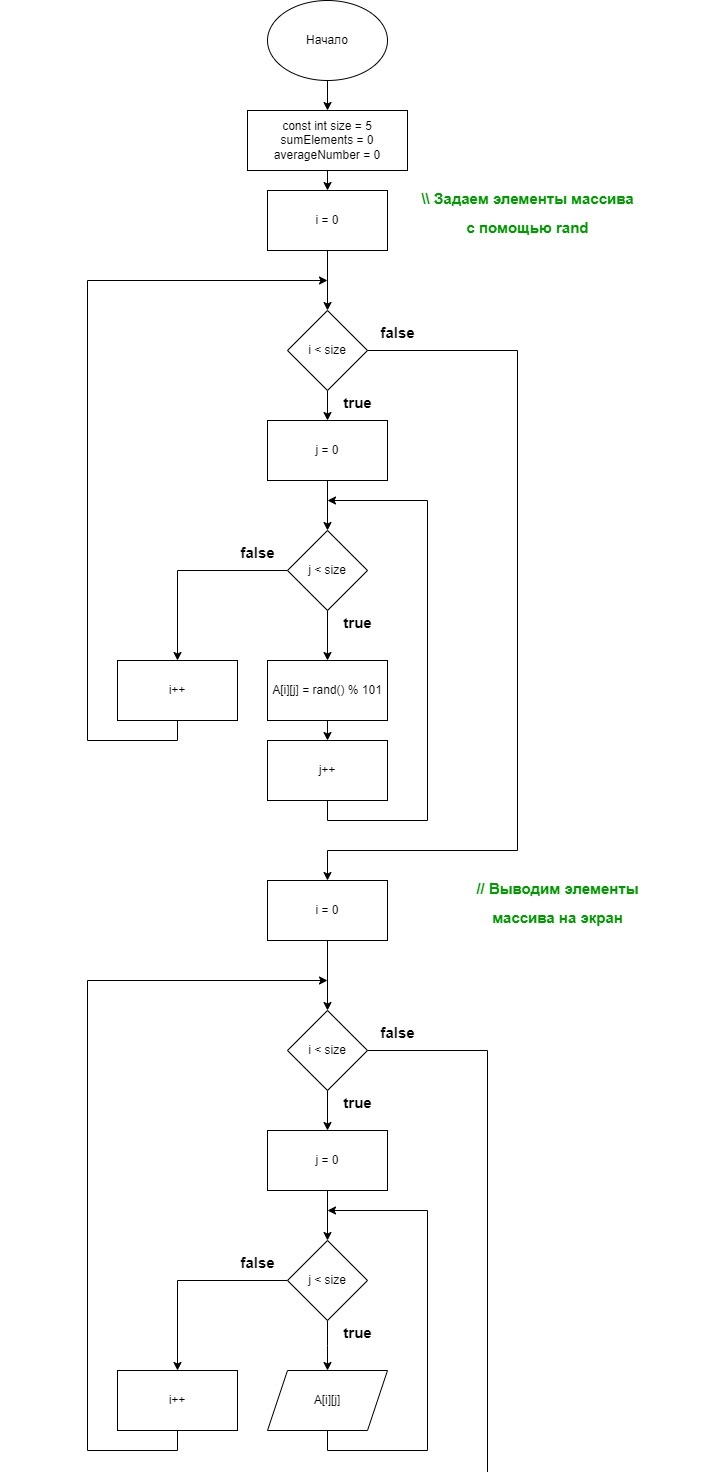
if (maxElement < A[i][j]) // если последующий элемент массива будет больше текущего, то maxElement примет новое значение

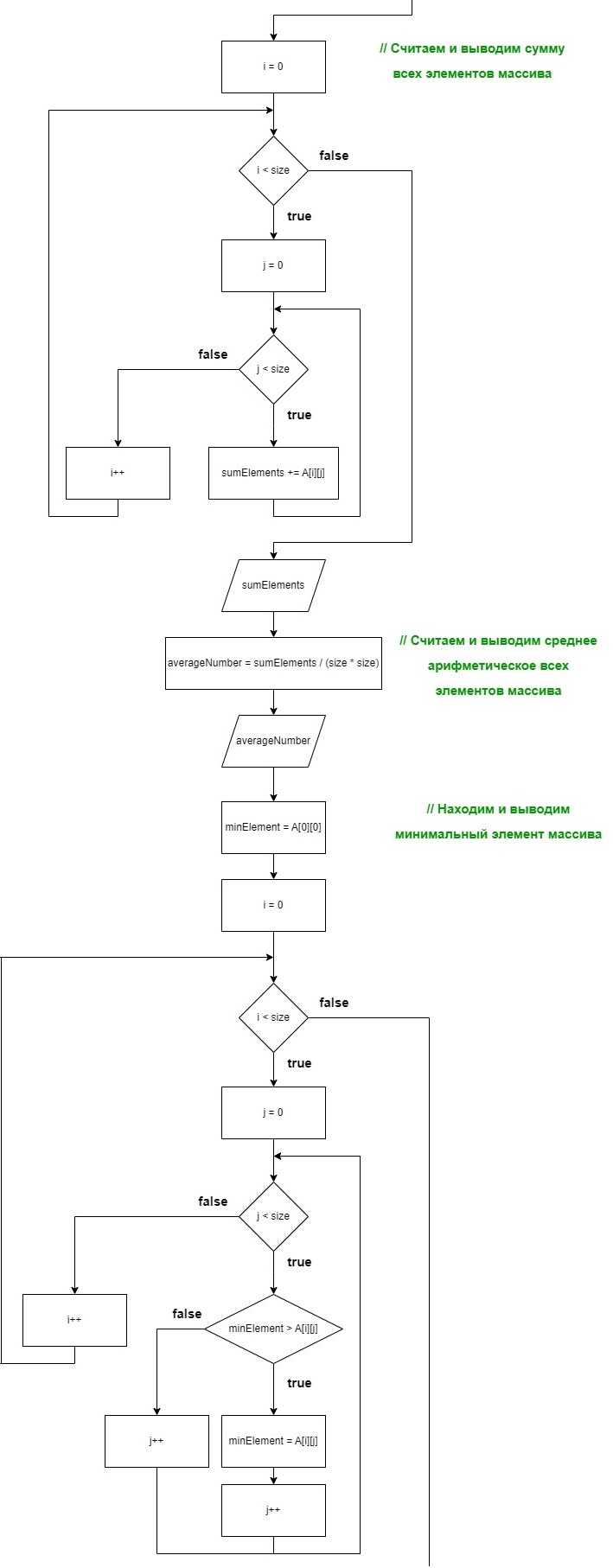
maxElement = A[i][j];

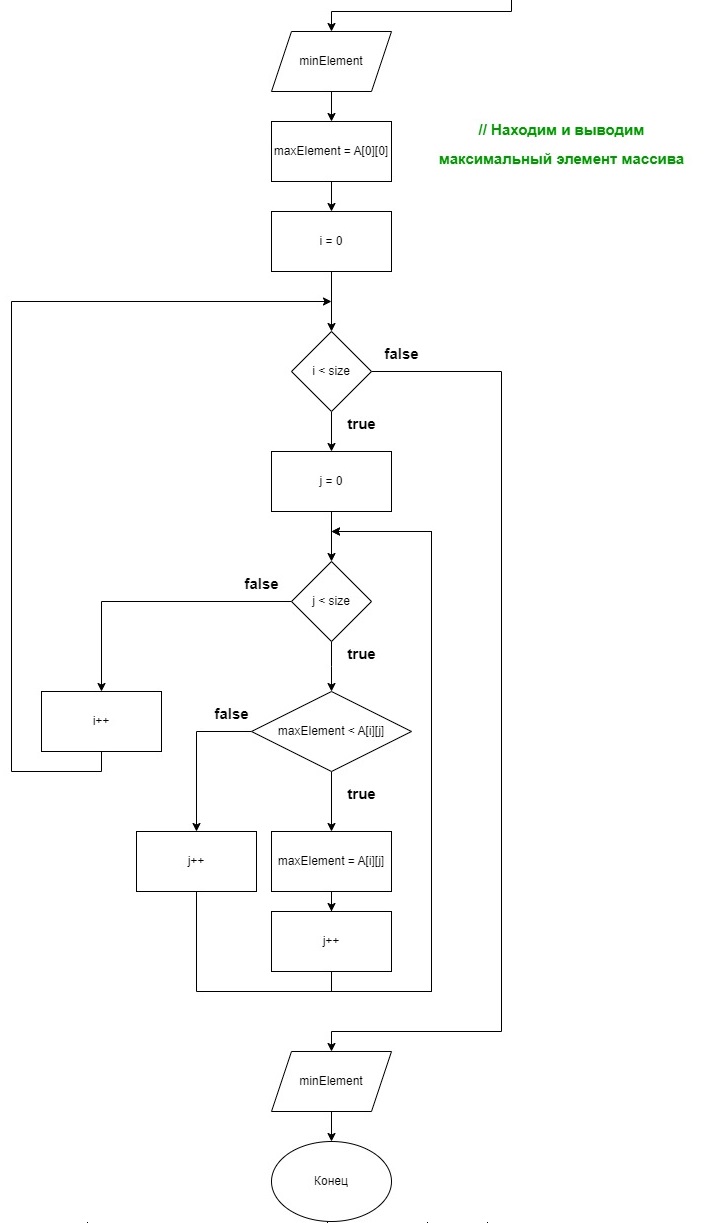
std::cout << "Максимальный элемент: " << maxElement << std::endl;

}

Блок-Схема:







### Задание 5:

В двумерном массиве целых чисел посчитать сумму элементов: в каждой строке; в каждом столбце; одновременно по всем строкам и всем столбцам. Оформить следующим образом:

3 5 6 7 | 21

12 1 1 1 | 15

0 7 12 1 | 20

---------------------

15 3 19 9 | 56

Решение:

#include <iostream>

#include <iomanip>

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(nullptr));

const int size = 5;

int A[size][size];

int sumRow[size] = {};

int sumColomn[size] = {};

int sum = 0;

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

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

A[i][j] = rand() % 101;

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

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

{

sumRow[i] += A[i][j];

sumColomn[i] += A[j][i];

sum += A[i][j];

}

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

{

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

{

std::cout << std::setw(4) << A[i][j] << " ";

}

std::cout << "| " << sumRow[i] << std::endl;

}

std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << std::endl;

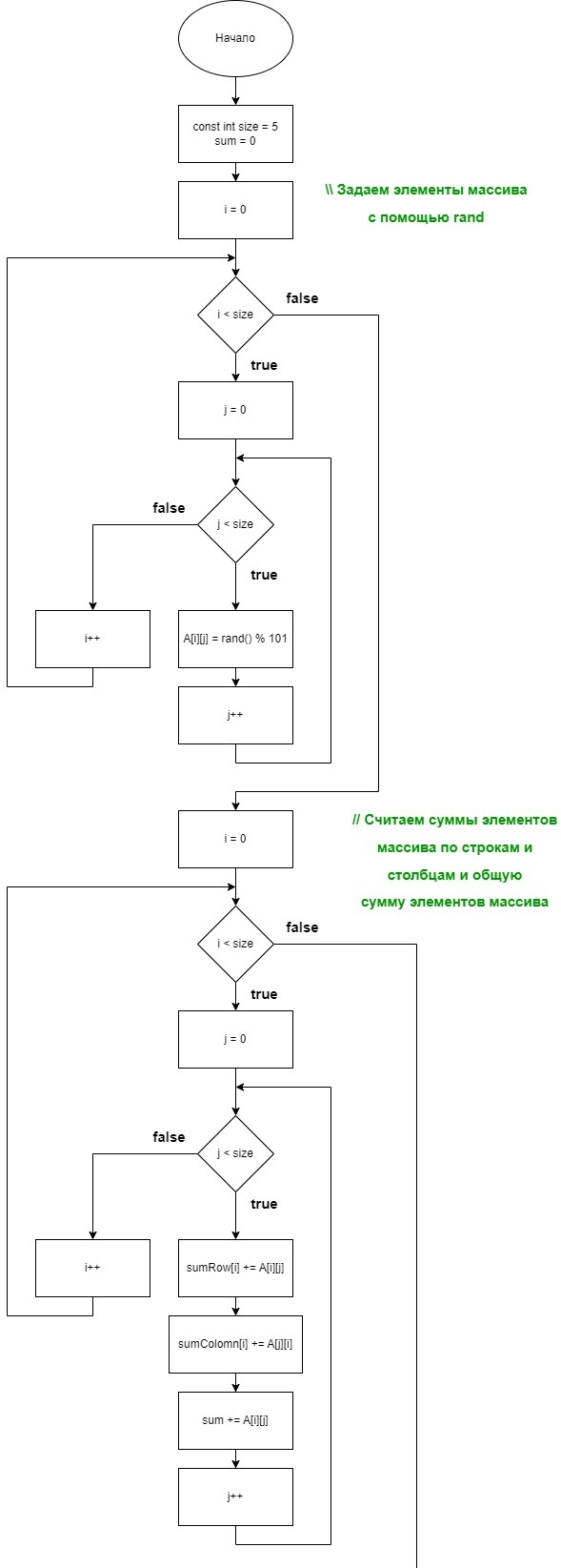
for (int i = 0; i < size; i++)

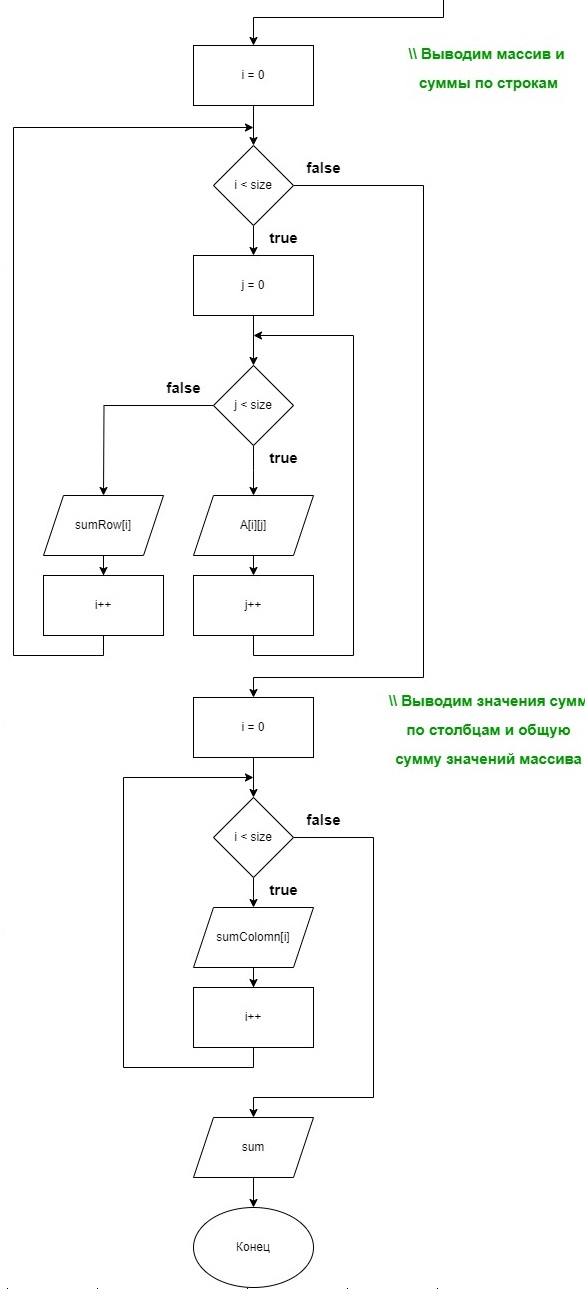
std::cout << std::setw(4) << sumColomn[i] << " ";

std::cout << "| " << sum;

}

Блок-Схема:





### Задание 6:

Напишите программу, в которой объявляется массив размером 5х10 и массив размером 5х5. Первый массив заполняется случайными числами, в диапазоне от 0 до 50. Второй массив заполняется по следующему принципу: первый элемент второго массива равен сумме первого и второго элемента первого массива, второй элемент второго массива равен сумму третьего и четвертого элемента первого массива

Решение:

#include <iostream>

#include <iomanip>

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(nullptr));

const int size = 5;

const int size2 = 10;

int A[size][size2];

int B[size][size];

std::cout << "\tМассив А:" << std::endl;

for (int i = 0; i < size; i++) // В первый массив 5х10 задаем значения с помощью rand и выводим на экран

{

for (int j = 0; j < size2; j++)

{

A[i][j] = rand() % 51;

std::cout << std::setw(3) << A[i][j] << " ";

}

std::cout << std::endl;

}

std::cout << std::endl;

std::cout << "\tМассив B:" << std::endl;

for (int i = 0; i < size; i++) // Создаем второй массив 5х5 из значений первого массива и выводим на экран

{

int x = 0; // пояснения см ниже, почему я задал эти переменные

int y = 1;

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

{

B[i][j] = A[i][j + x] + A[i][j + y];

x++;

y++;

std::cout << std::setw(3) << B[i][j] << " ";

}

std::cout << std::endl;

}

std::cout << std::endl;

/\*

Пояcнения по переменным x и y:

При i=0, j=0 : b00 = а00 + а01 -> a0(0+0) + a0(0+1) -> ai(j+0) + ai(j+1) -> пусть х=0, у=1

При i=0, j=1 : b01 = a02 + a03 -> a0(1+1) + a0(1+2) -> ai(j+1) + ai(j+2) -> пусть х=1, у=2

При i=0, j=2 : b02 = a04 + a05 -> a0(2+2) + a0(2+3) -> ai(j+2) + ai(j+3) -> пусть х=2, у=3

При i=0, j=3 : b03 = a06 + a07 -> a0(3+3) + a0(3+4) -> ai(j+3) + ai(j+4) -> пусть х=3, у=4

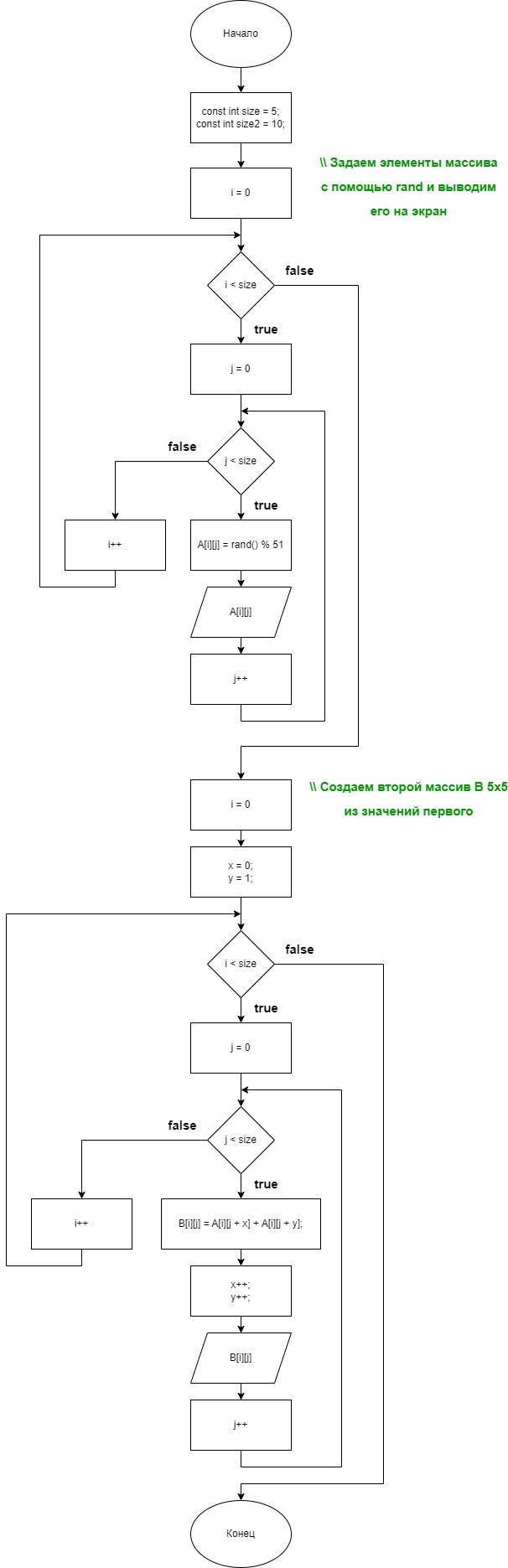
При i=0, j=4 : b04 = a08 + a09 -> a0(4+5) + a0(4+6) -> ai(j+4) + ai(j+5) -> пусть х=4, у=5

Мы видим закономерность, что после каждой итерации в цикле j мы должны делать х++ и у++. А начальный х и у задать 0 и 1 соответственно.

\*/

}

Блок-Схема:



## **По теме: «Функции».**

### Задание 1:

Написать функцию, которая принимает два параметра: основание степени и показатель

степени, и вычисляет степень числа на основе полученных данных.

### Решение:

#include <iostream>

int raiseToThePower(int x, int y);

int main()

{

setlocale(LC\_ALL, "Russian");

int x, y;

std::cout << "Введите основание степени - > ";

std::cin >> x;

std::cout << "Введите степень - > ";

std::cin >> y;

raiseToThePower(x, y);

}

int raiseToThePower(int x, int y)

{

int answer = x ;

for (int i = 2; i <= y; i++) // первая итерация i=2, так как х\*х это уже 2я степень

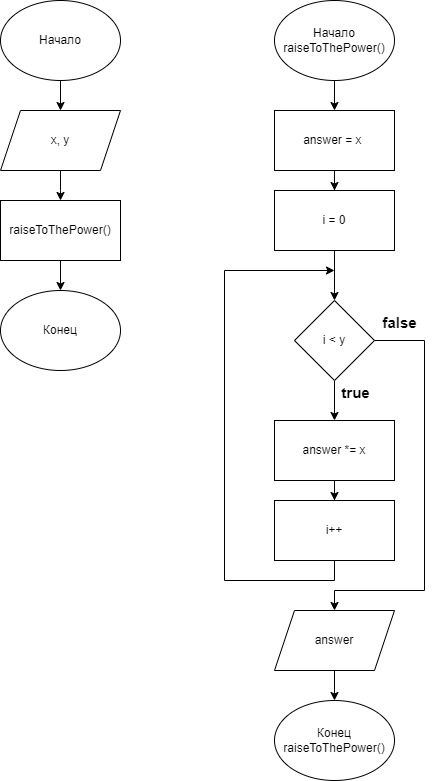
answer \*= x;

std::cout << x << " в степени " << y << " равно " << answer;

return answer;

}

Блок-Схема:



### Задание 2:

Написать функцию, которая получает в качестве параметров 2 целых числа и возвращает сумму

чисел из диапазона между ними.

### Решение:

#include <iostream>

int sumNumbers(int x, int y);

int main()

{

setlocale(LC\_ALL, "Rus");

int x, y;

std::cout << "Введите два целых числа x и y" << std::endl;

std::cout << "x - > ";

std::cin >> x;

std::cout << "y - > ";

std::cin >> y;

sumNumbers (x, y);

}

int sumNumbers(int x, int y)

{

int sum = 0;

for (int i = x; i <= y; i++)

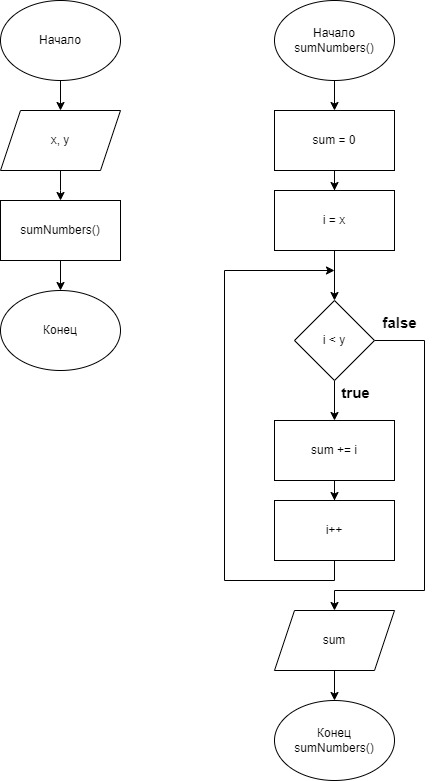
sum += i;

std::cout << "Сумма чисел в диапазоне от " << x << " до " << y << " равна - > " << sum << std::endl;

return sum;

}

Блок-Схема:



### Задание 3:

Число называется совершенным, если сумма всех его делителей равна ему самому. Напишите

функцию поиска таких чисел во введенном интервале.

Решение:

#include <iostream>

void perfectNumber(int x, int y);

int main()

{

setlocale(LC\_ALL, "rus");

long long int x, y;

std::cout << "Введите два натуральных числа x и y" << std::endl;

std::cout << "x - > ";

std::cin >> x;

std::cout << std::endl;

std::cout << "y - > ";

std::cin >> y;

std::cout << std::endl;

perfectNumber(x, y);

}

void perfectNumber(int x, int y)

{

int temp; // временная переменная, если x>y

if (x > y)

{

temp = x;

x = y;

y = temp;

}

std::cout << "Совершенные числа из диапазона от " << x << " до " << y << " :" << std::endl;

for (int i = x; i <= y; i++)

{

long long int sumDiv = 0;

for (int div = 1; div <= i/2; div++) // div <= i/2 ,так как все целые делители числа не больше половины этого числа

{

if (i % div == 0)

sumDiv += div;

}

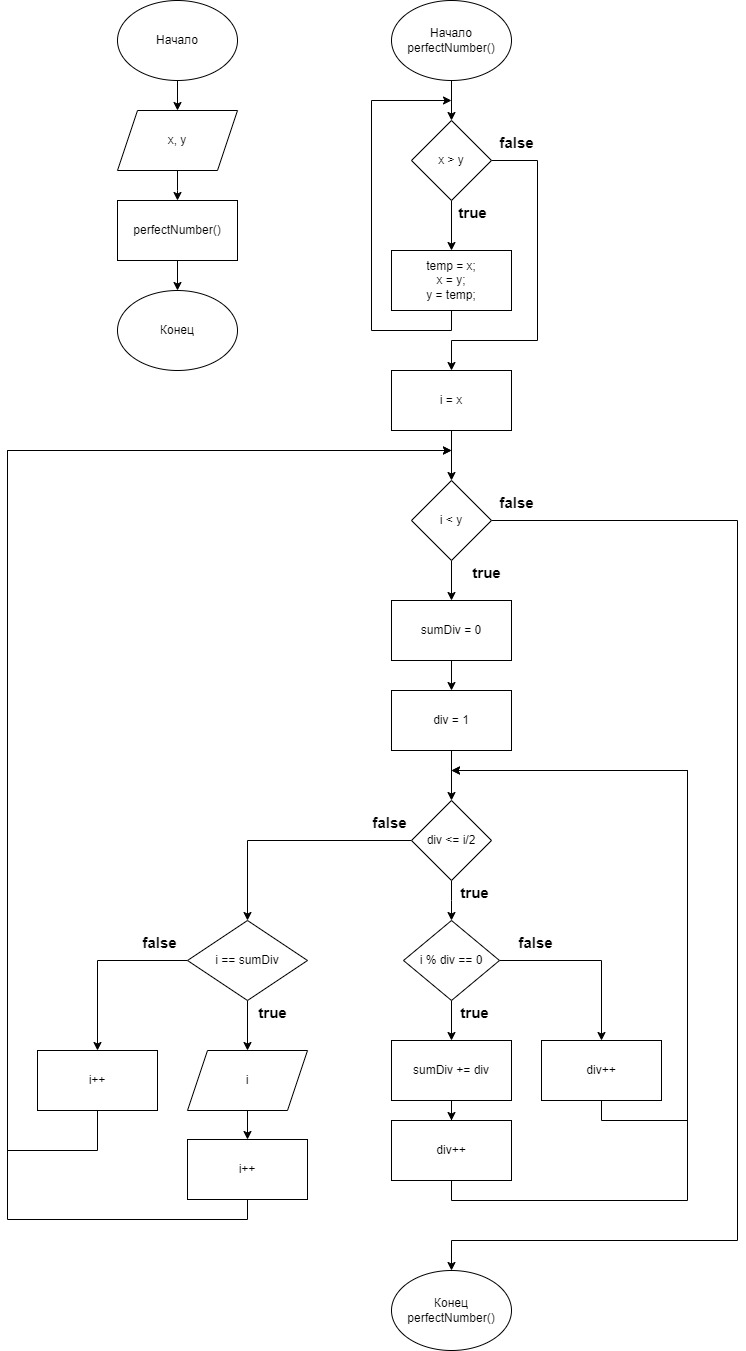
if (i == sumDiv)

std::cout << i << std::endl;

}

}

Блок-Схема:



### Задание 4:

Написать функцию, выводящую на экран переданную ей игральную карту.

Решение:

#include <iostream>

#include <iomanip>

void card(int suit, int num);

int main()

{

setlocale(LC\_ALL, "Rus");

int a, s;

std::cout << "Выберите карту:\n1 - Туз \n2 - Двойка\n3 - Тройка\n4 - Четверка\n5 - Пять\n6 - Шесть\n7 - Семь\n8 - Восемь\n9 - Девять\n10 - Десять\n11 - Валет\n12 - Дама\n13 - Король" << std::endl;

std::cin >> a;

system("cls");

std::cout << "Выберите масть карты:\n1. Черви\n2. Бубы\n3. Крести\n4. Пики\n";

std::cin >> s;

system("cls");

card(s, a);

}

void card(int suit, int num)

{

char card[13] = { 'A','2','3','4','5','6','7','8','9','10','J','Q','K' };

std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(5) << card[num - 1] << std::setw(15) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

switch (suit)

{

case 1:

std::cout << "|" << std::setw(12) << "Черви" << std::setw(8) << "|" << std::endl;

break;

case 2:

std::cout << "|" << std::setw(12) << "Бубы" << std::setw(8) << "|" << std::endl;

break;

case 3:

std::cout << "|" << std::setw(12) << "Крести" << std::setw(8) << "|" << std::endl;

break;

case 4:

std::cout << "|" << std::setw(12) << "Пики" << std::setw(8) << "|" << std::endl;

break;

}

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(15) << card[num - 1] << std::setw(5) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "|" << std::setw(20) << "|" << std::endl;

std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << std::endl;

}

### Задание 5:

Написать функцию, которая определяет, является ли «счастливым» шестизначное число.

Решение:

#include <iostream>

void happyNumber(int sixNumber);

int main()

{

setlocale(LC\_ALL, "Russian");

std::cout << "Счастливое число (англ. happy number) — число, определённое следующим процессом: начиная с любого \nположительного целого числа, мы заменяем это число суммой квадратов его цифр в десятичной \nсистеме счисления и повторяем данный процесс, пока число либо не станет равно 1 \n(где весь процесс остановится), или попадёт в бесконечный цикл, не содержащий 1. \nЧисла, для которых данный процесс заканчивается единицей, называются счастливыми числами, \nв то время как те, для которых процесс не заканчивается единицей, считаются \nнесчастливыми числами (или грустными числами)." << std::endl;

std::cout << std::endl;

std::cout << "К примеру, число 19 счастливое, так как соответствующая ему последовательность такова: \n1^2 + 9^2 = 82 \n8^2 + 2^2 = 68 \n6^2 + 8^2 = 100 \n1^2 + 0^2 + 0^2 = 1." << std::endl;

std::cout << std::endl;

int sixNumber;

std::cout << "Введите 6-ти значное число -> " << std::endl;

std::cin >> sixNumber;

while (sixNumber < 99999 or sixNumber > 999999) //Данный фильтр выставил согласно заданию. Его лучше убрать, чтобы свериться в правильности ответов с википедией и водить числа до 1000.

{

std::cout << "Ошибка! Введите 6-ти значное число -> " << std::endl;

std::cin >> sixNumber;

}

happyNumber(sixNumber);

}

void happyNumber(int sixNumber)

{

int x = sixNumber;

int sum = 0;

int count = 0;

do

{

sum = 0;

for (int i = 0; i < 6; i++) // данный цикл разбивает 6ти значное число на 6 отдельных чисел

{

int c = sixNumber % 10;

sixNumber /= 10;

c \*= c; // возводит в квадрат

sum += c; // и собирает сумму квадратов

}

sixNumber = sum; // Новое значение сравниваем с 1 и дальше разбиваем числ, если sum != 1.

count++; // счетчик, чтобы остановить бесконечный цикл и сделать вывод, что число НЕ счастливое

} while (sum != 1 and count != 100);

if (sum == 1)

std::cout << "Число " << x << " является счастливым" << std::endl;

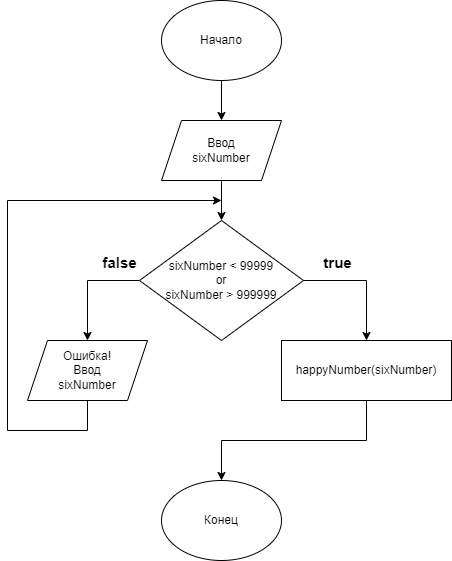
else

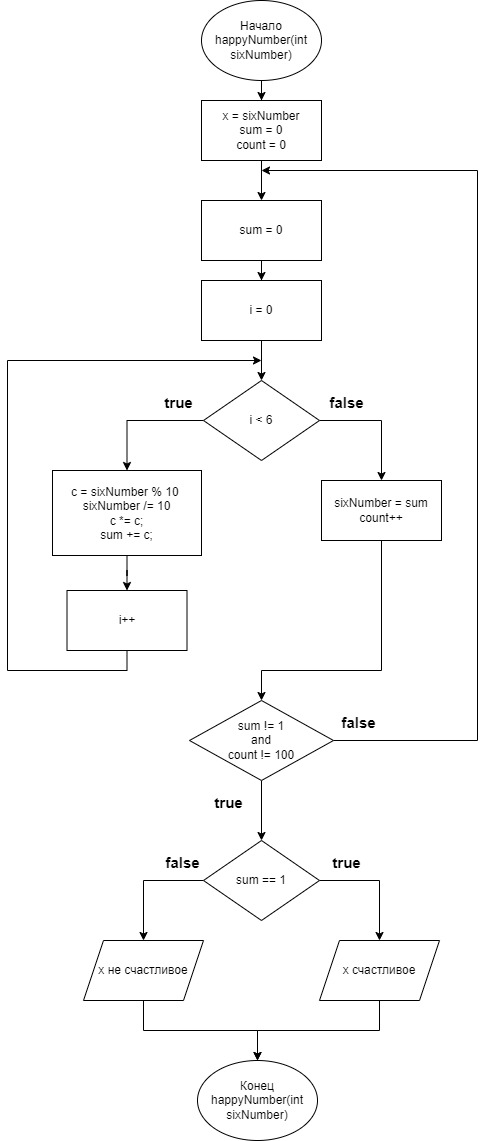
std::cout << "Число " << x << " НЕ является счастливым" << std::endl;

// P.S. Надеюсь правильно понял задание и сделал все верно.

}

Блок-Схема:





### Задание 6:

Написать функцию, которая принимает две даты (т.е. функция принимает шесть параметров) и вычисляет разность в днях между этими датами. Для решения этой задачи необходимо также написать функцию, которая определяет, является ли год високосным.

Решение:

#include <iostream>

bool leapYear(int y); // проверка високосный год или нет

void difDays(int firstDay, int secondDay, int firstMonth, int secondMonth, int firstYear, int secondYear); // разница в днях

int firstYear, firstMonth, firstDay; // введем глобальные переменные для всех функций.

int secondYear, secondMonth, secondDay;

int main()

{

setlocale(LC\_ALL, "rus");

// Для начала пользователь вводит первую и вторую дату. Программа проверяет, правильно ли пользователь вводит даты и просит ввести заново, если дата введена неверно.

// Например, пользователь не сможет ввести 29 февраля 2023 года, тк год не високосный.

std::cout << "\tВведите первую дату:" << std::endl;

std::cout << "Год -> ";

std::cin >> firstYear;

std::cout << std::endl;

do {

std::cout << "Месяц -> ";

std::cin >> firstMonth;

} while (firstMonth < 1 or firstMonth > 12);

std::cout << std::endl;

if (leapYear(firstYear)) // високосный год или нет

{

if (firstMonth == 1 or firstMonth == 3 or firstMonth == 5 or firstMonth == 7 or firstMonth == 8 or firstMonth == 10 or firstMonth == 12) // 31 день

do {

std::cout << "День -> ";

std::cin >> firstDay;

} while (firstDay < 1 or firstDay > 31);

else if (firstMonth == 2) // 29 дней

do {

std::cout << "День -> ";

std::cin >> firstDay;

} while (firstDay < 1 or firstDay > 29);

else if (firstMonth == 4 or firstMonth == 6 or firstMonth == 9 or firstMonth == 11) // 30 дней

do {

std::cout << "День -> ";

std::cin >> firstDay;

} while (firstDay < 1 or firstDay > 30);

}

else

{

if (firstMonth == 1 or firstMonth == 3 or firstMonth == 5 or firstMonth == 7 or firstMonth == 8 or firstMonth == 10 or firstMonth == 12) // 31 день

do {

std::cout << "День -> ";

std::cin >> firstDay;

} while (firstDay < 1 or firstDay > 31);

else if (firstMonth == 2) // 28 дней

do {

std::cout << "День -> ";

std::cin >> firstDay;

} while (firstDay < 1 or firstDay > 28);

else if (firstMonth == 4 or firstMonth == 6 or firstMonth == 9 or firstMonth == 11) // 30 дней

do {

std::cout << "День -> ";

std::cin >> firstDay;

} while (firstDay < 1 or firstDay > 30);

}

system("cls");

std::cout << "\tПервая дата -> " << firstDay << "." << firstMonth << "." << firstYear << std::endl;

std::cout << std::endl;

std::cout << "\tВведите вторую дату:" << std::endl;

std::cout << "Год -> ";

std::cin >> secondYear;

std::cout << std::endl;

do {

std::cout << "Месяц -> ";

std::cin >> secondMonth;

} while (secondMonth < 1 or secondMonth > 12);

std::cout << std::endl;

if (leapYear(secondYear)) // високосный год или нет

{

if (secondMonth == 1 or secondMonth == 3 or secondMonth == 5 or secondMonth == 7 or secondMonth == 8 or secondMonth == 10 or secondMonth == 12) // 31 день

do {

std::cout << "День -> ";

std::cin >> secondDay;

} while (secondDay < 1 or secondDay > 31);

else if (secondMonth == 2) // 29 дней

do {

std::cout << "День -> ";

std::cin >> secondDay;

} while (secondDay < 1 or secondDay > 29);

else if (secondMonth == 4 or secondMonth == 6 or secondMonth == 9 or secondMonth == 11) // 30 дней

do {

std::cout << "День -> ";

std::cin >> secondDay;

} while (secondDay < 1 or secondDay > 30);

}

else

{

if (secondMonth == 1 or secondMonth == 3 or secondMonth == 5 or secondMonth == 7 or secondMonth == 8 or secondMonth == 10 or secondMonth == 12) // 31 день

do {

std::cout << "День -> ";

std::cin >> secondDay;

} while (secondDay < 1 or secondDay > 31);

else if (secondMonth == 2) // 28 дней

do {

std::cout << "День -> ";

std::cin >> secondDay;

} while (secondDay < 1 or secondDay > 28);

else if (secondMonth == 4 or secondMonth == 6 or secondMonth == 9 or secondMonth == 11) // 30 дней

do {

std::cout << "День -> ";

std::cin >> secondDay;

} while (secondDay < 1 or secondDay > 30);

}

system("cls");

std::cout << std::endl;

// Выводит даты на экран.

std::cout << "\tПервая дата -> " << firstDay << "." << firstMonth << "." << firstYear << std::endl;

std::cout << "\tВторая дата -> " << secondDay << "." << secondMonth << "." << secondYear << std::endl;

std::cout << std::endl;

difDays(firstDay, secondDay, firstMonth, secondMonth, firstYear, secondYear);

}

bool leapYear(int y)

{

if (y % 400 == 0) return true;

if (y % 100 == 0) return false;

if (y % 4 == 0) return true;

return false;

}

void difDays(int firstDay, int secondDay, int firstMonth, int secondMonth, int firstYear, int secondYear)

{

/\*

Логика вычислений:

1) Пусть программа полностью посчитает дни ОТ 1го числа 1го месяца введенного первого года (firstYear) ДО 31го числа 12го месяца года перед введенным последним годом (secondYear).

То есть дни последнего года не берем в расчет.

Получим общее кол-во дней за полные года в заданном периоде, не включая последний год.

2) Посчитаем сколько дней прошло с 1го месяца включительно до введенного месяца первой даты. Получим minus\_FirstPart\_FirstYear.

3) Посчитаем сколько дней прошло с 1го месяца включительно до введенного месяца второй даты. Получим plus\_FirstPart\_LastYear.

4) Посчитаем сколько дней прошло с 1го дня включительно до введенного дня выбранного месяца первой даты. Получим minus\_FirstPart\_FisrtMonth.

5) Посчитаем сколько дней прошло с 1го дня включительно до введенного дня выбранного месяца второй даты. Получим plus\_FirstPart\_LastMonth.

\*/

int days{ 0 };

for (int i = firstYear; i < secondYear; i++)

{

if (leapYear(i))

days += 366;

else days += 365;

}

int minus\_FirstPart\_FirstYear = 0;

for (int i = 1; i < firstMonth; i++)

{

if (leapYear(firstYear))

{

if (i == 1 or i == 3 or i == 5 or i == 7 or i == 8 or i == 10 or i == 12)

minus\_FirstPart\_FirstYear += 31;

if (i == 2)

minus\_FirstPart\_FirstYear += 29;

if (i == 4 or i == 6 or i == 9 or i == 11)

minus\_FirstPart\_FirstYear += 30;

}

else

{

if (i == 1 or i == 3 or i == 5 or i == 7 or i == 8 or i == 10 or i == 12)

minus\_FirstPart\_FirstYear += 31;

if (i == 2)

minus\_FirstPart\_FirstYear += 28;

if (i == 4 or i == 6 or i == 9 or i == 11)

minus\_FirstPart\_FirstYear += 30;

}

}

int plus\_FirstPart\_LastYear = 0;

for (int i = 1; i < secondMonth; i++)

{

if (leapYear(secondYear))

{

if (i == 1 or i == 3 or i == 5 or i == 7 or i == 8 or i == 10 or i == 12)

plus\_FirstPart\_LastYear += 31;

if (i == 2)

plus\_FirstPart\_LastYear += 29;

if (i == 4 or i == 6 or i == 9 or i == 11)

plus\_FirstPart\_LastYear += 30;

}

else

{

if (i == 1 or i == 3 or i == 5 or i == 7 or i == 8 or i == 10 or i == 12)

plus\_FirstPart\_LastYear += 31;

if (i == 2)

plus\_FirstPart\_LastYear += 28;

if (i == 4 or i == 6 or i == 9 or i == 11)

plus\_FirstPart\_LastYear += 30;

}

}

int minus\_FirstPart\_FisrtMonth = 0;

for (int d = 0; d < firstDay; d++)

minus\_FirstPart\_FisrtMonth++;

int plus\_FirstPart\_LastMonth = 0;

for (int d = 0; d < secondDay; d++)

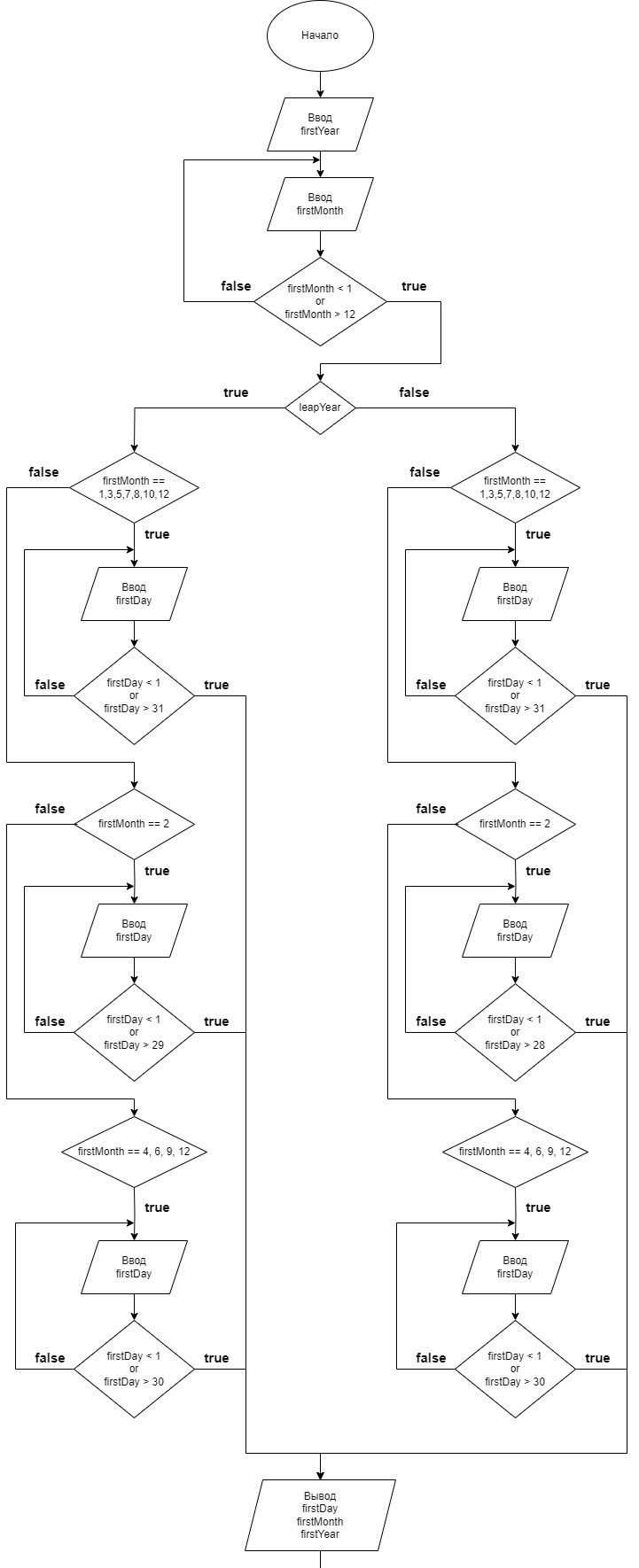
plus\_FirstPart\_LastMonth++;

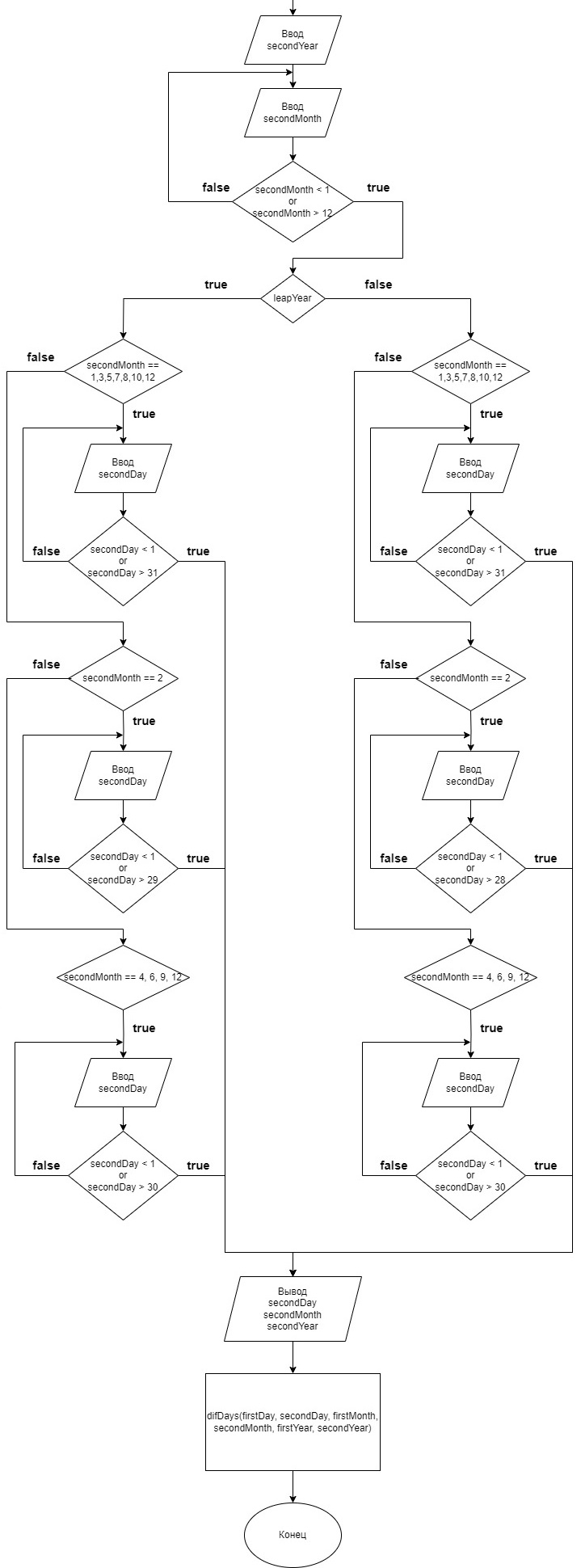
days = days - minus\_FirstPart\_FirstYear - minus\_FirstPart\_FisrtMonth + plus\_FirstPart\_LastYear + plus\_FirstPart\_LastMonth;

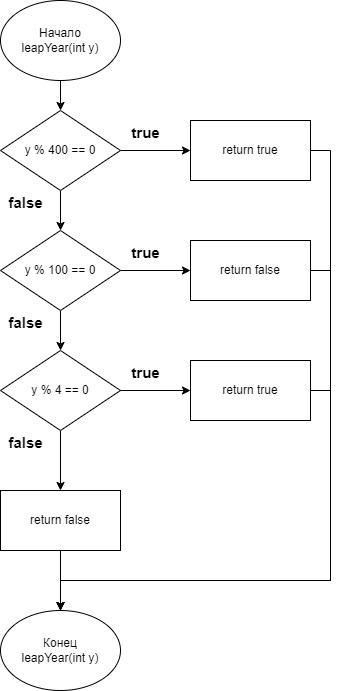
std::cout << "Разница в количестве дней -> " << days;

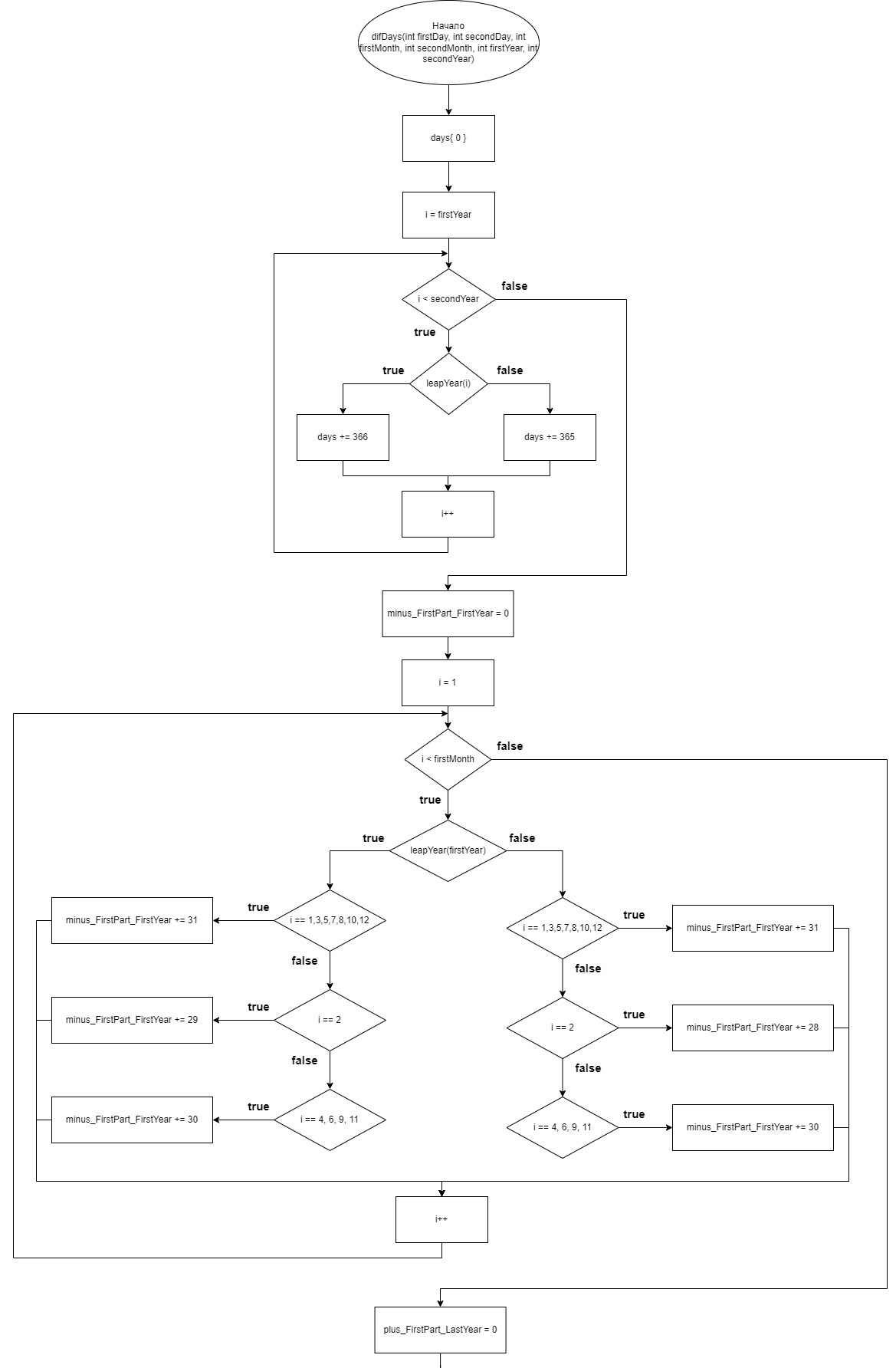
}

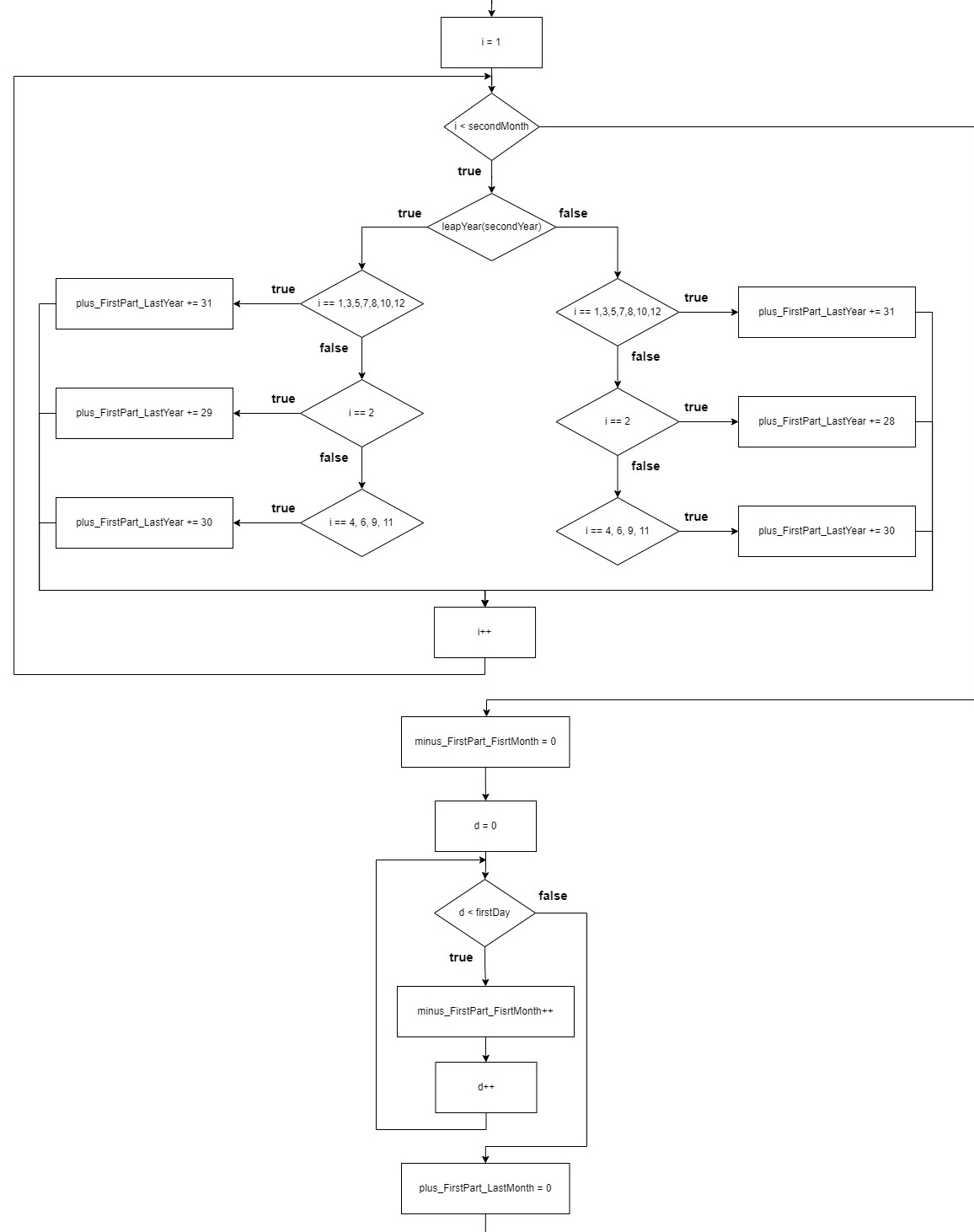
Блок-Схема:

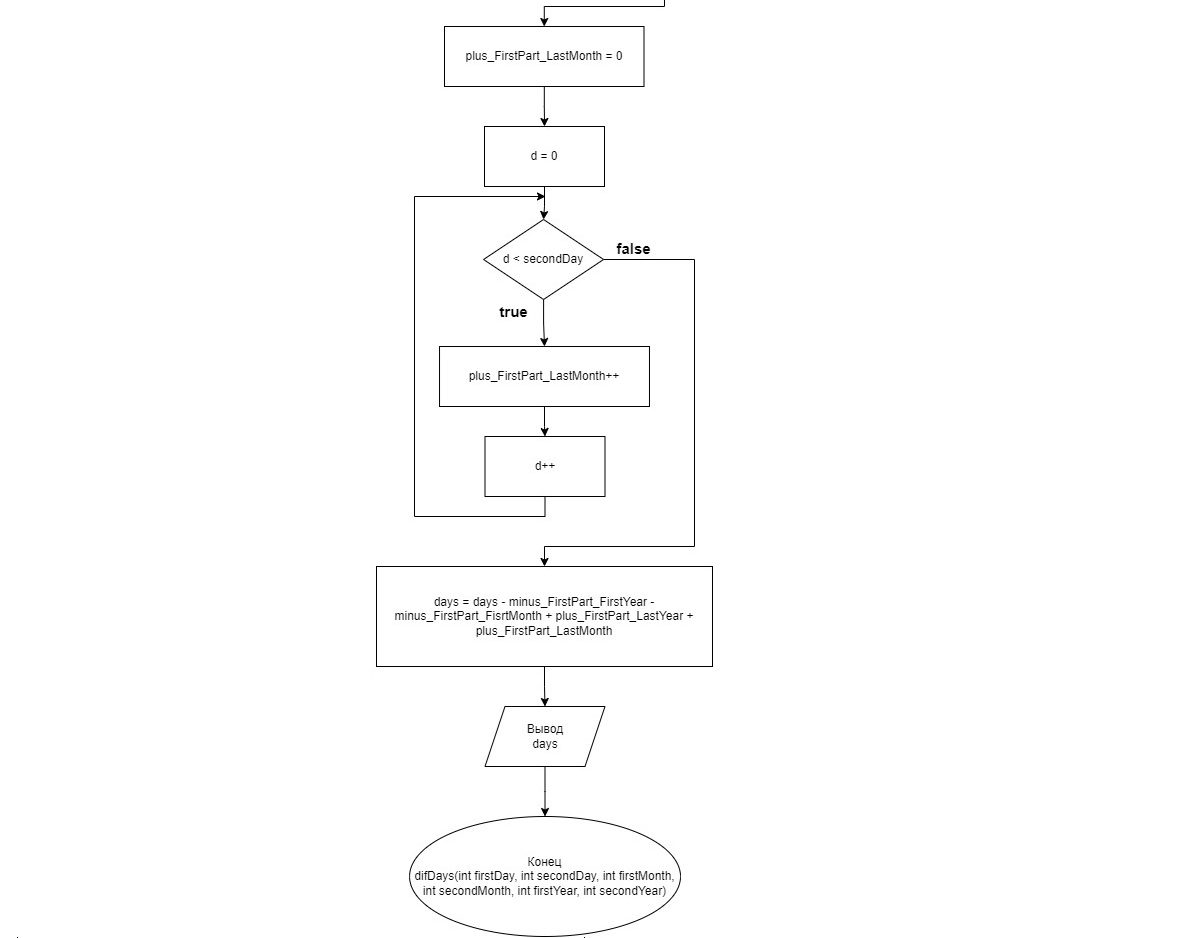












### Задание 7:

Написать функцию, определяющую среднее арифметическое элементов передаевамого ей массива.

Решение:

#include <iostream>

#include <iomanip>

const int size (5);

void anvrNum(int A[][size], int size); //прототип функции

int main()

{

setlocale(LC\_ALL, "Rus");

srand(time(nullptr));

int x;

int A[size][size];

std::cout << "\tВыберите способ передачи элементов в массив А: " << std::endl;

std::cout << "1 - задать элементы массива рандомом от -25 до 25. " << std::endl;

std::cout << "2 - задать элементы массива самосоятельно. " << std::endl;

std::cin >> x;

switch (x)

{

case 1:

{

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

A[i][j] = rand() % 51 + (-25);

break;

}

case 2:

{

std::cout << "Введите числа массива " << size << "\*" << size << " :" << std::endl;

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

{

std::cout << i + 1 << " строка, " << j + 1 << " столбец -> ";

std::cin >> A[i][j];

}

break;

}

}

std::cout << "\tМассив А:" << std::endl;

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

{

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

{

std::cout << std::setw(5) << A[i][j];

}

std::cout << std::endl;

}

anvrNum(A, size);

}

void anvrNum(int A[][size], int size)

{

int sum{};

int quantity{};

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

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

{

sum += A[i][j];

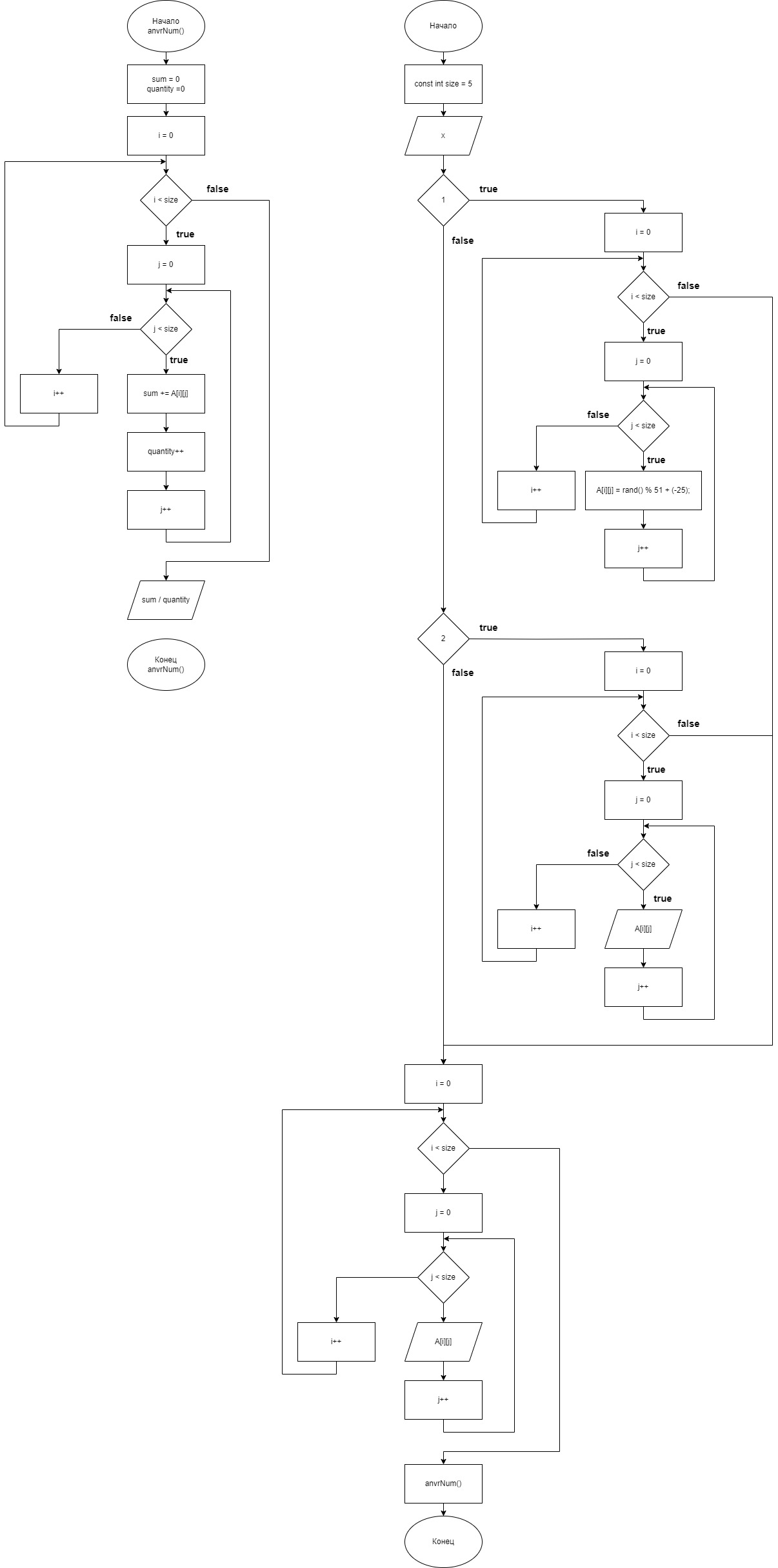
quantity++;

}

std::cout << "Среднее арифметическое элементов передаваемого ей массива - > " << (double)sum / (double)quantity;

}

Блок-Схема:



### Задание 8:

Написать функцию, определяющую количество положительных, отрицательных и нулевых элементов передаваемого ей массива.

Решение:

#include <iostream>

#include <iomanip>

const int size{5};

void positiveNumbers(int A[][size], int size);

void negativeNumbers(int A[][size], int size);

void zeroNumbers(int A[][size], int size);

int main()

{

setlocale(LC\_ALL, "Rus");

srand(time(nullptr));

int x;

int A[size][size];

std::cout << "\tВыберите способ передачи элементов в массив А: " << std::endl;

std::cout << "1 - задать элементы массива рандомом от -25 до 25. " << std::endl;

std::cout << "2 - задать элементы массива самосоятельно. " << std::endl;

std::cin >> x;

switch (x)

{

case 1:

{

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

A[i][j] = rand() % 51 + (-25);

break;

}

case 2:

{

std::cout << "Введите числа массива " << size << "\*" << size << " :" << std::endl;

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

{

std::cout << i + 1 << " строка, " << j + 1 << " столбец -> ";

std::cin >> A[i][j];

}

break;

}

}

std::cout << "\tМассив А:" << std::endl;

for (int i{ 0 }; i < size; i++)

{

for (int j{ 0 }; j < size; j++)

{

std::cout << std::setw(5) << A[i][j];

}

std::cout << std::endl;

}

positiveNumbers(A, size);

negativeNumbers(A, size);

zeroNumbers(A, size);

}

void positiveNumbers(int A[][size], int size)

{

int positiveCount{ 0 };

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

{

if (A[i][j] > 0)

positiveCount++;

}

std::cout << "Количество положительных элементов в массиве А - > " << positiveCount << std::endl;

}

void negativeNumbers(int A[][size], int size)

{

int negativeCount{ 0 };

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

{

if (A[i][j] < 0)

negativeCount++;

}

std::cout << "Количество отрицательных элементов в массиве А - > " << negativeCount << std::endl;

}

void zeroNumbers(int A[][size], int size)

{

int zeroCount{ 0 };

for (int i{ 0 }; i < size; i++)

for (int j{ 0 }; j < size; j++)

{

if (A[i][j] == 0)

zeroCount++;

}

std::cout << "Количество нулевых элементов в массиве А - > " << zeroCount << std::endl;

}

Блок-Схема:

