**Задание 1:**

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

степени, и вычисляет степень числа на основе полученных данных.

#include <iostream>

using namespace std;

double Calculate (double base, int degree) { //объявляем функцию, которая принимает два параметра для вычисления степени числа– основание и степень

double result = 1.0; //объявление переменной, которая хранит результат возведения числа в степень, первоначальное значение 1

if (degree >= 0) { //если показатель больше или равен 0

for (int i = 0; i < degree; i++) { // количество итераций от 0 до значения степени

result \*= base; // результат, равен перемножению оснований в количестве значений введённой степени, если степень = 0, то результат равен 1

}

} else { //иначе

for (int i = 0; i < -degree; i++) { //если у нас степень меньше 0, то количество итераций от 0 до значения степени, знак минус делает из отрицательного значения степени положительное для правильной записи цикла for

result /= base; // результат – единица делённая на основание (по количеству итераций)

}

}

return result; // возвращаемая переменная у функции - result

}

int main() {

setlocale(LC\_ALL, "ru");

//введение переменных

double base;

int degree;

cout << "Введите основание степени: ";

cin >> base;

cout << "Введите показатель степени: ";

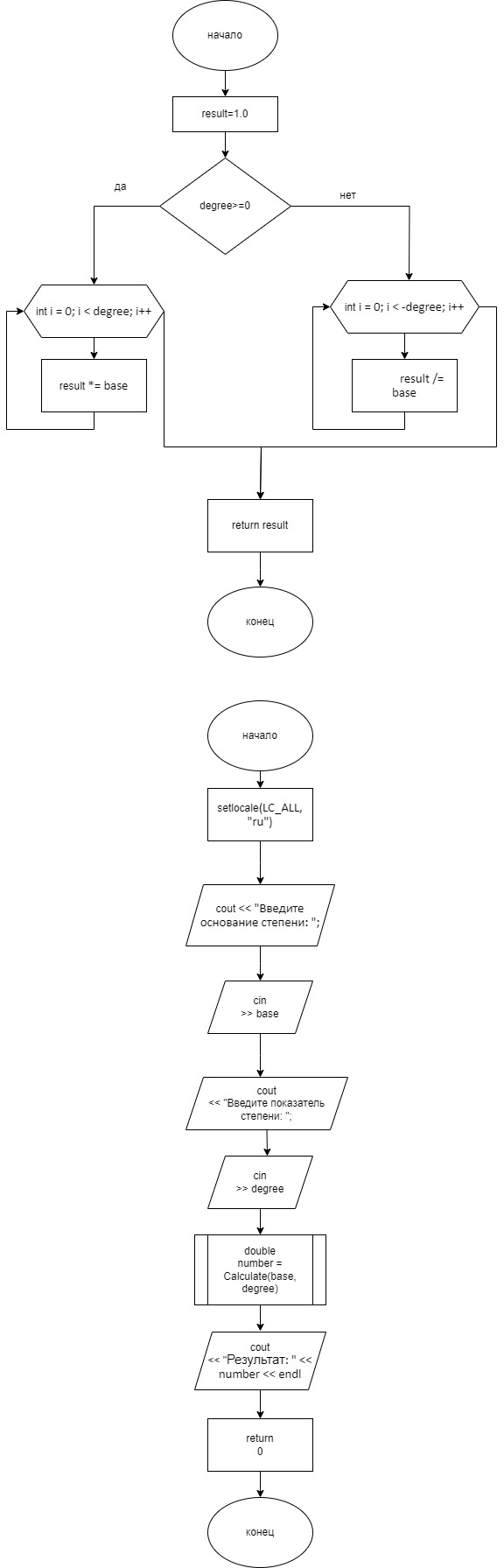
cin >> degree;

double number = Calculate(base, degree); //вызов функции Calculate для переменной number

cout << "Результат: " << number << endl;

return 0;

}



**Задание 2**

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

чисел из диапазона между ними.

#include <iostream>

using namespace std;

int RangeSum(int first, int second) { //объявляем функцию, которая принимает два параметра для вычисления cуммы диапазона- первое число и второе

int sum = 0; //переменная итоговой суммы, первоначальное значение 0

if (first < second) { //если первое число меньше второго

for (int i = first + 1; i < second; i++) { //проход по циклу

sum += i; //прибавляем к переменной суммы 1

}

} else { //иначе

for (int i = second + 1; i < first; i++) { //проход по циклу

sum += i; //прибавляем к переменной суммы 1

}

}

return sum; //возвращаемая переменная функции - sum

}

int main() {

setlocale(LC\_ALL, "ru");

//ввод переменных

int num1, num2;

cout << "Введите два целых числа: ";

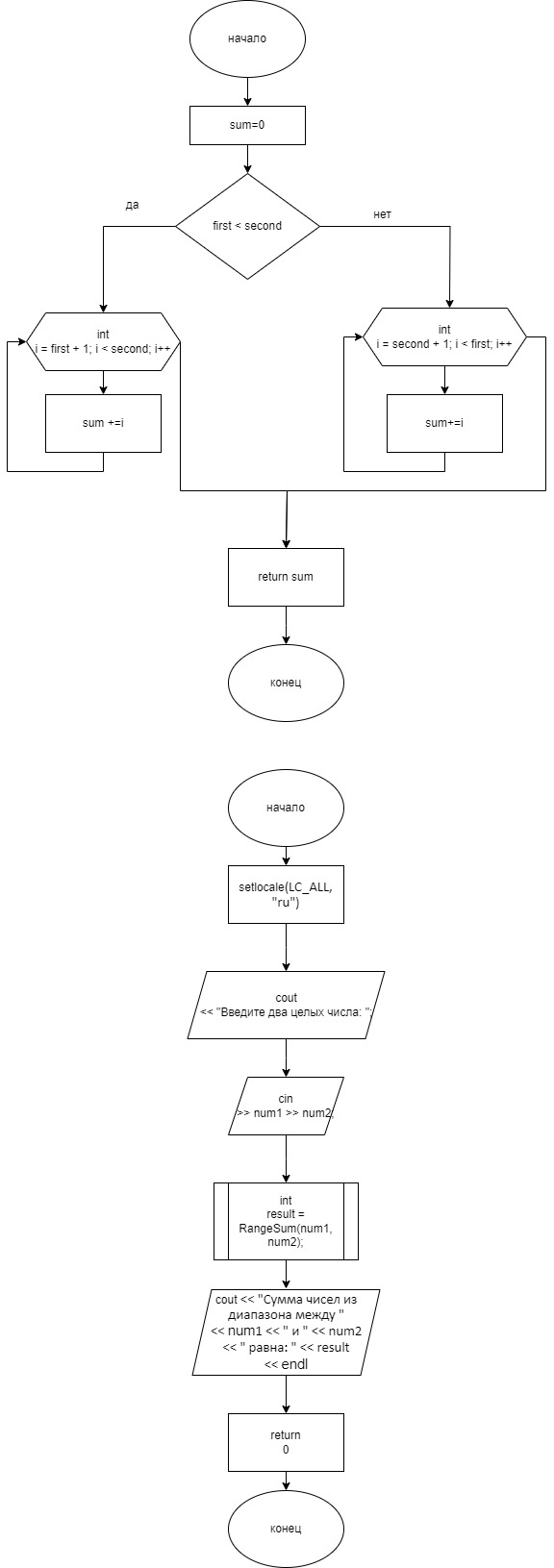
cin >> num1 >> num2;

int result = RangeSum(num1, num2); //вызов функции RangeSum для переменной result

cout << "Сумма чисел из диапазона между " << num1 << " и " << num2 << " равна: " << result << endl;

return 0;

}



**Задание 3**

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

функцию поиска таких чисел во введенном интервале

#include <iostream>

using namespace std;

bool Perfectnum (int num) { //объявляем функцию, которая принимает параметр – число для определения совершенное оно или нет

int sum = 0; // переменная суммы, первоначальное значение 0

for (int i = 1; i <= num/2; i++) { //Из теории чисел известно, что все делители произвольного числа меньше половины этого числа

if (num % i == 0) { // если число делится на счётчик цикла без остатка

sum += i; // прибавляем к сумме счётчик

}

}

return sum == num; // возвращаемая переменная функции – sum, равная num

}

void PerfectNumbers(int start, int end) { //объявляем функцию, которая принимает параметры- начало и конец диапазона, для вывода всех совершенных чисел

cout << "Perfect numbers in the interval [" << start << ", " << end << "] are: ";

for (int i = start; i <= end; i++) {

if (Perfectnum (i)) { //вызываем функцию определения совершенного числа для счётчика i

std::cout << i << " ";

}

}

std::cout << std::endl;

}

int main() {

setlocale(LC\_ALL, "ru");

int start, end;

cout << "Enter the start of the interval: ";

cin >> start;

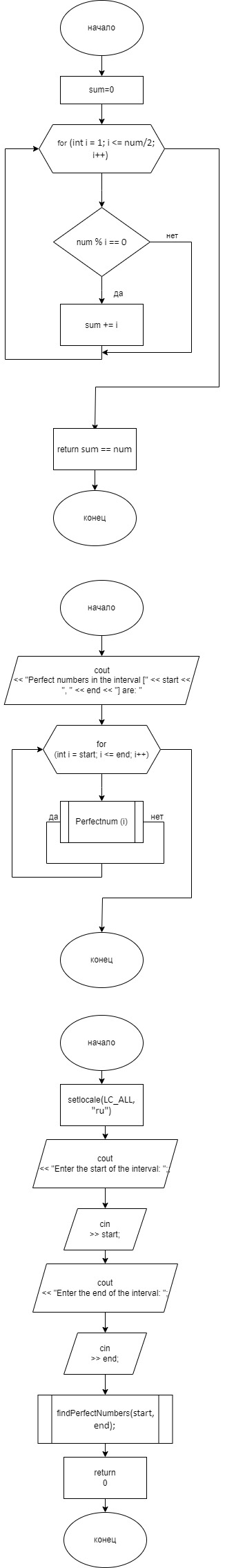
cout << "Enter the end of the interval: ";

cin >> end;

PerfectNumbers (start, end); //вызов функции для нахождения совершенных чисел в диапазоне чисел, введённых пользователем

return 0;

}



**Задание 4**

Написать функцию, выводящую на экран переданную ей игральную карту.

#include <iostream>

#include <string>

using namespace std;

void BackCard(string rank, string suit) { //объявляем функцию, которая принимает параметры – ранг карты и масть

cout << rank << " of " << suit << endl;

}

int main() {

setlocale(LC\_ALL, "ru");

string rank;

string suit;

cout << "Введите ранг карты: ";

cin >> rank;

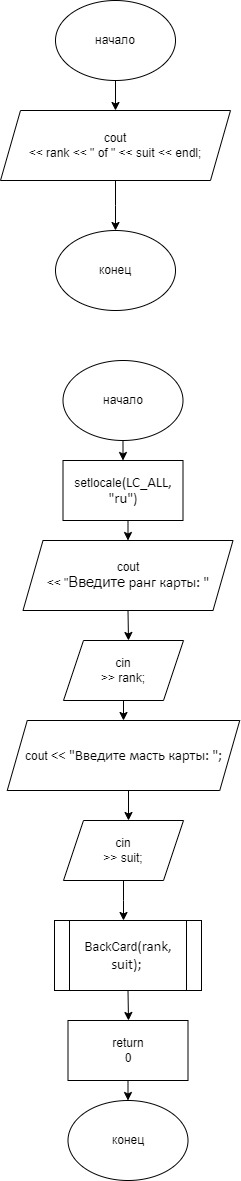
cout << "Введите масть карты: ";

cin >> suit;

BackCard(rank, suit); //объявляем функцию для выведения ранга и масти карты

return 0;

}



**Задание 5**

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

Счастливое шестизначное число - это число, сумма первых трех цифр которых равна сумме трех последних цифр.

#include <iostream>

using namespace std;

bool HappyNumber(int number) { //объявляем функцию, которая принимает параметр – шестизначное число для определения счастливое оно или нет

int sum\_first\_half = 0; //сумма первых трёх чисел

int sum\_second\_half = 0; //сумма последних трёх чисел

for (int i = 0; i < 6; i++) { //проход по каждой цифре числа

if (i < 3) { //первые три цифры

sum\_first\_half += number % 10; //суммируем их остатки от деления на 10

} else { //иначе

sum\_second\_half += number % 10; //суммируем остатки трёх последних цифр от деления на 10

}

}

return sum\_first\_half == sum\_second\_half; //возвращаем равенство сумм трёх первых цифр и последних

}

int main() {

setlocale(LC\_ALL, "ru");

int num;

cout << "Введите шестизначное число: ";

cin >> num;

if (HappyNumber(num)) { //объявляем функцию для проверки условия на счастливое число

cout << num << " счастливое число." << endl;

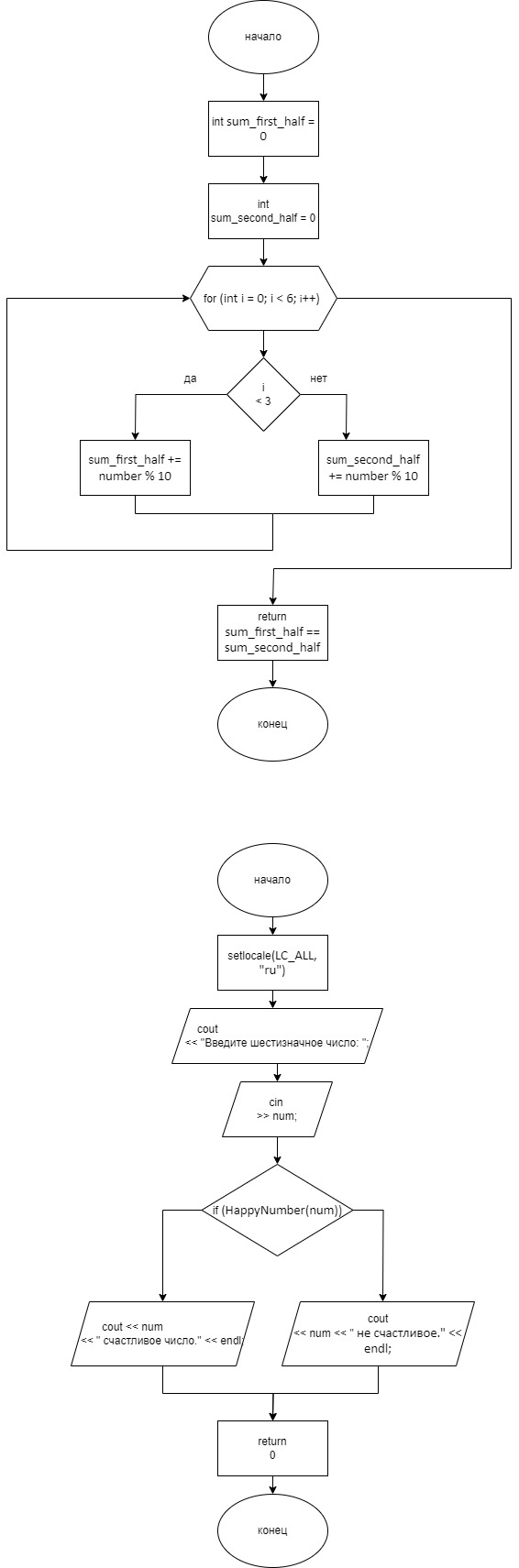
} else {

cout << num << " не счастливое." << endl;

}

return 0;

}



**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

#include <iostream>;

using namespace std;

bool isLeapYear(int year) { //объявляем функцию для определения високосного года, параметр - год

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) { //условие високосного года – год должен либо кратен 4, но при этом не кратен 100, либо кратен 400.

return true;

} else {

return false;

}

}

int daysBetweenDates (int day1, int month1, int year1, int day2, int month2, int year2) {

int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; //количество дней в месяцах

int days = 0;

// Посчитаем дни между первым годом и вторым

for (int year = year1; year < year2; year++) {

if (isLeapYear(year)) { //если год високосный

days += 366; //количество дней 366

} else { //если нет

days += 365; //365 дней

}

}

**Задание 7**

Написать функцию, определяющую среднее арифметическое элементов передаевамого ей

массива.

#include <iostream>

using namespace std;

// Функция для вычисления среднего арифметического элементов массива

double average(int arr[], int size) {

int sum = 0; // Сумма элементов массива

// Цикл для суммирования элементов массива

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

sum += arr[i];

}

return (double)sum / size; // Возвращаем среднее арифметическое элементов массива

}

int main() {

setlocale(LC\_ALL, "ru")

// Вводим массив элементов

int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

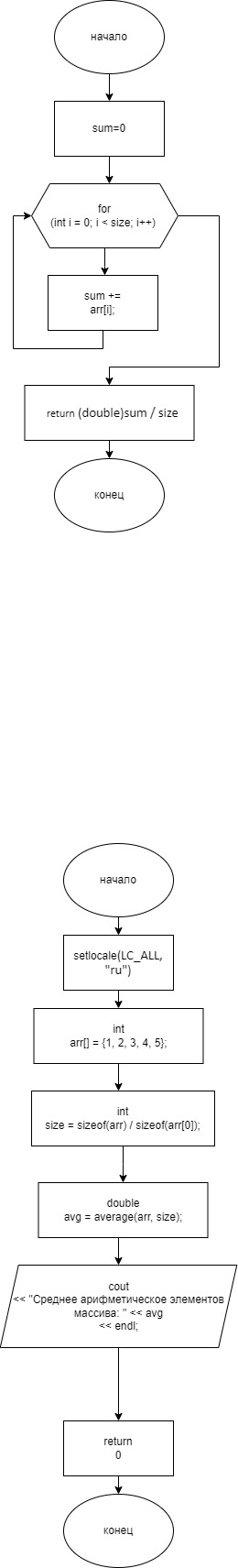
// Вычисляем среднее арифметическое элементов массива через функцию

double avg = average(arr, size);

cout << "Среднее арифметическое элементов массива: " << avg << endl;

return 0;

}



**Задание 8**

Написать функцию, определяющую количество положительных, отрицательных и нулевых

элементов передаваемого ей массива.

#include <iostream>

using namespace std;

// Функция для подсчета количества положительных, отрицательных и нулевых элементов в массиве

tuple<int, int, int> count\_positives\_negatives\_zeros(int arr[], int size) {

// Количество положительных, отрицательных и нулевых элементов

int positive\_count = 0;

int negative\_count = 0;

int zero\_count = 0;

// Цикл для подсчета количества элементов каждого типа

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

if (arr[i] > 0) {

positive\_count++;

} else if (arr[i] < 0) {

negative\_count++;

} else {

zero\_count++;

}

}

// Возвращаем кортеж с количеством элементов каждого типа

return make\_tuple(positive\_count, negative\_count, zero\_count);

}

int main() {

setlocale(LC\_ALL, "ru");

// Вводим массив элементов

int arr[] = {1, -2, 0, 3, -4, 5, 0};

int size = sizeof(arr) / sizeof(arr[0]);

// Подсчитываем количество элементов каждого типа

int positive\_count, negative\_count, zero\_count;

tie(positive\_count, negative\_count, zero\_count) = count\_positives\_negatives\_zeros(arr, size);

// Выводим результат

cout << "Количество положительных элементов: " << positive\_count << endl;

cout << "Количество отрицательных элементов: " << negative\_count << endl;

cout << "Количество нулевых элементов: " << zero\_count << endl;

return 0;

}

