Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №7.2.**

**«Функции с переменным числом параметров»**

Выполнил студент гр. РИС-24-1б

Чижов Денис Николаевич

Проверил:

Доц. Каф. ИТАС 

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

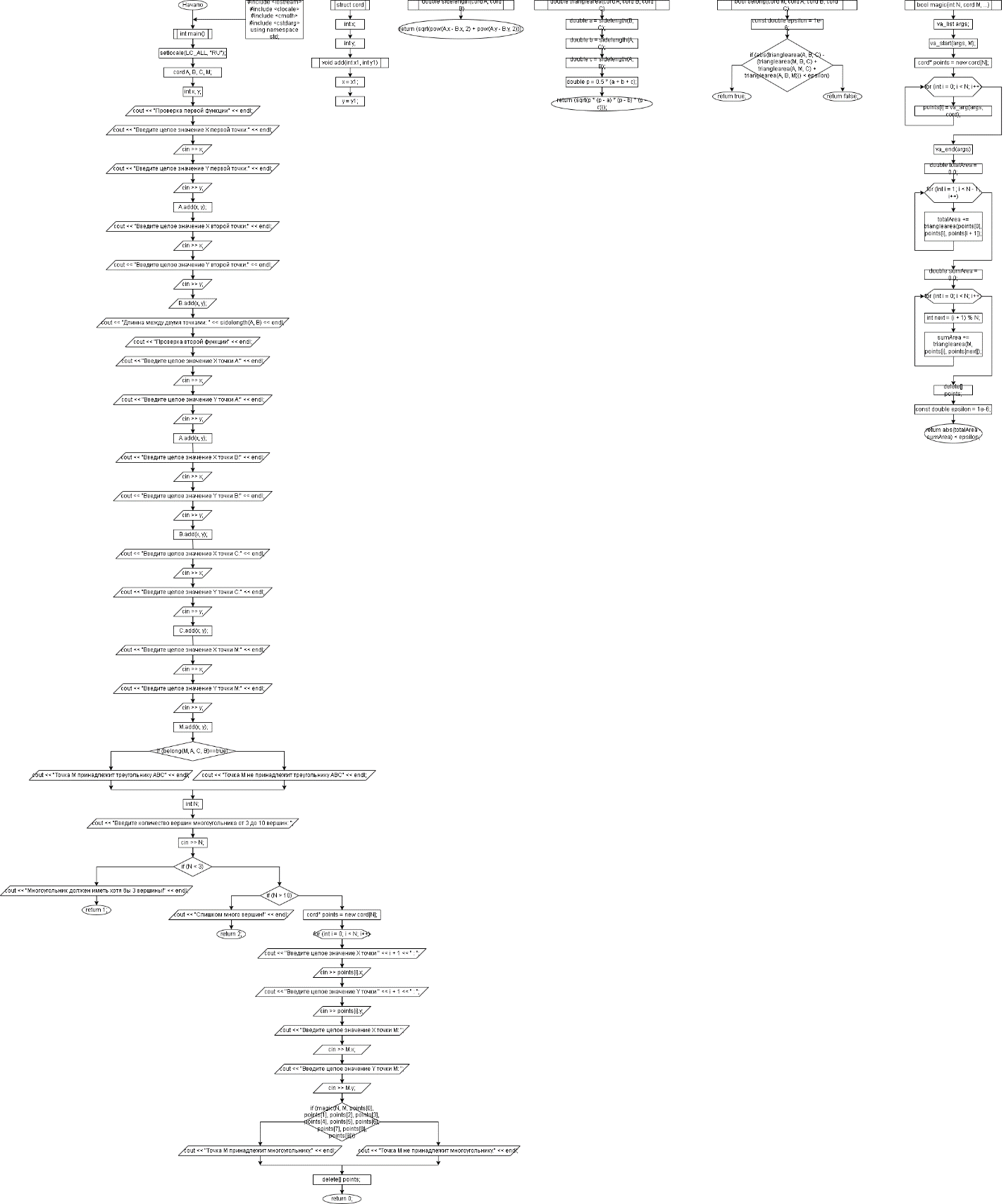
**Задача:**

Написать функцию (или макроопределение), которая находит длину стороны по координатам его точек. Написать функцию belong, которая определяет принадлежит ли точка М с координатами (х , у) треугольнику, заданному координатами вершин. Написать функцию c переменным числом параметров, которая определяет принадлежит ли точка М выпуклому многоугольнику, заданному координатами своих вершин.

**Анализ:**

Программа находит длину между двумя точками по теореме Пифагора. Функция belong сравнивает площадь треугольника ABC с суммой треугольников ABM, BCM, CAM. Функция magic разбивает многоугольник на треугольники и сравнивает площадь многоугольника с сумой треугольников с точкой M.

**Блок-схема:**



**Код:**

#include <iostream>

#include <clocale>

#include <cmath>

#include <cstdarg> // библиотека для работы с функциями с переменным числом аргументов

using namespace std;

struct cord {

int x;

int y;

void add(int x1, int y1) {

x = x1;

y = y1;

}

};

double sidelength(cord A, cord B) { //функция, которая находит длину стороны по координатам его точек

return (sqrt(pow(A.x - B.x, 2) + pow(A.y - B.y, 2)));

}

double trianglearea(cord A, cord B, cord C){ //функция счёта площади треугольника

double a = sidelength(B, C);

double b = sidelength(A, C);

double c = sidelength(A, B);

double p = 0.5 \* (a + b + c);

return (sqrt(p \* (p - a) \* (p - b) \* (p - c)));

}

// Функция для определения, принадлежит ли точка M треугольнику ABC

bool belong(cord M, cord A, cord B, cord C) {

// Сравниваем площади с учетом погрешности

const double epsilon = 1e-6;

if (abs(trianglearea(A, B, C) - (trianglearea(M, B, C) + trianglearea(A, M, C) + trianglearea(A, B, M))) < epsilon) {

return true;

}

else {

return false;

}

}

bool magic(int N, cord M, ...) {

va\_list args;

va\_start(args, M);

// Создаем массив для хранения вершин многоугольника

cord\* points = new cord[N];

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

points[i] = va\_arg(args, cord);

}

va\_end(args);

// Вычисляем площадь многоугольника

double totalArea = 0.0;

for (int i = 1; i < N - 1; i++) {

totalArea += trianglearea(points[0], points[i], points[i + 1]);

}

//totalArea += (trianglearea(points[N - 1], points[N - 2], points[0]));

// Вычисляем сумму площадей треугольников, образованных точкой M и каждой парой соседних вершин

double sumArea = 0.0;

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

int next = (i + 1) % N; // Следующая вершина (с учетом замыкания на первую)

sumArea += trianglearea(M, points[i], points[next]);

}

delete[] points; // Освобождаем память

// Сравниваем площади с учетом погрешности

const double epsilon = 1e-6;

return abs(totalArea - sumArea) < epsilon;

}

int main()

{

setlocale(LC\_ALL, "RU");

cord A, B, C, M;

int x, y;

cout << "Проверка первой функции" << endl;

cout << "Введите целое значение X первой точки:" << endl;

cin >> x;

cout << "Введите целое значение Y первой точки:" << endl;

cin >> y;

A.add(x, y);

cout << "Введите целое значение X второй точки:" << endl;

cin >> x;

cout << "Введите целое значение Y второй точки:" << endl;

cin >> y;

B.add(x, y);

cout << "Длинна между двумя точками: " << sidelength(A, B) << endl;

cout << "Проверка второй функции" << endl;

cout << "Введите целое значение X точки A:" << endl;

cin >> x;

cout << "Введите целое значение Y точки A:" << endl;

cin >> y;

A.add(x, y);

cout << "Введите целое значение X точки B:" << endl;

cin >> x;

cout << "Введите целое значение Y точки B:" << endl;

cin >> y;

B.add(x, y);

cout << "Введите целое значение X точки C:" << endl;

cin >> x;

cout << "Введите целое значение Y точки C:" << endl;

cin >> y;

C.add(x, y);

cout << "Введите целое значение X точки M:" << endl;

cin >> x;

cout << "Введите целое значение Y точки M:" << endl;

cin >> y;

M.add(x, y);

if (belong(M, A, C, B)==true) {

cout << "Точка M принадлежит треугольнику ABC" << endl;

}

else {

cout << "Точка M не принадлежит треугольнику ABC" << endl;

}

int N;

cout << "Введите количество вершин многоугольника от 3 до 10 вершин: ";

cin >> N;

if (N < 3) {

cout << "Многоугольник должен иметь хотя бы 3 вершины!" << endl;

return 1;

}

if (N > 10) {

cout << "Слишком много вершин!" << endl;

return 2;

}

// Создаем массив для хранения вершин многоугольника

cord\* points = new cord[N];

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

cout << "Введите целое значение X точки " << i + 1 << " : ";

cin >> points[i].x;

cout << "Введите целое значение Y точки " << i + 1 << " : ";

cin >> points[i].y;

}

cout << "Введите целое значение X точки M: ";

cin >> M.x;

cout << "Введите целое значение Y точки M: ";

cin >> M.y;

// Вызов функции magic для проверки принадлежности точки M многоугольнику

if (magic(N, M, points[0], points[1], points[2], points[3], points[4], points[5], points[6], points[7], points[8], points[9])) {

cout << "Точка M принадлежит многоугольнику." << endl;

}

else {

cout << "Точка M не принадлежит многоугольнику." << endl;

}

delete[] points; // Освобождаем память

return 0;

}

**Вывод:**

Программа работает. Задача решена.