Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**ЛАБОРАТОРНАЯ РАБОТА**

«Лаб. 7.2 - Функция с переменным количеством параметров»

**Выполнил:**

студент группы РИС-23-2б

Борисов Никита Андреевич

**Проверила:**

доцент кафедры ИТАС

О.А. Полякова

Пермь, 2024 г.

**Задача:**

Решить указанную в варианте задачу, используя функции с

переменным числом параметров.

По варианту 24 написать функцию (или макроопределение), которая находит длину стороны по координатам его точек. Написать функцию, которая вычисляет площадь треугольника, заданного координатами вершин. Написать функцию c переменным числом параметров, которая определяет площадь выпуклого многоугольника, заданного координатами своих вершин.

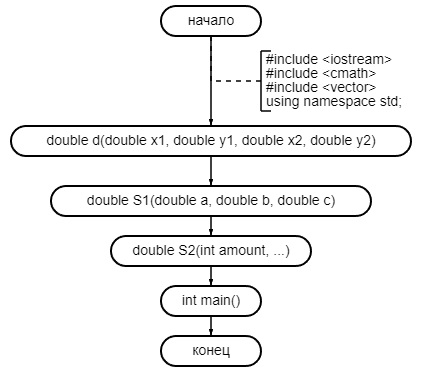
**Анализ:**

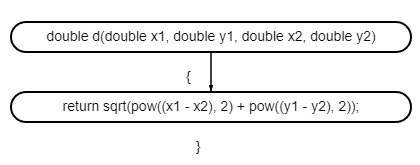
Задача заключается в написании функций с переменным числом параметров для решения следующих задач:

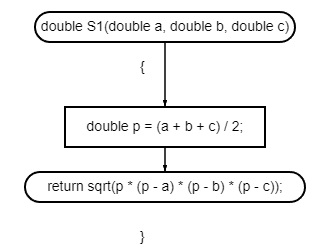
* Нахождение длины стороны по координатам ее точек.
* Вычисление площади треугольника, заданного координатами его вершин.
* Определение площади выпуклого многоугольника, заданного координатами его вершин.

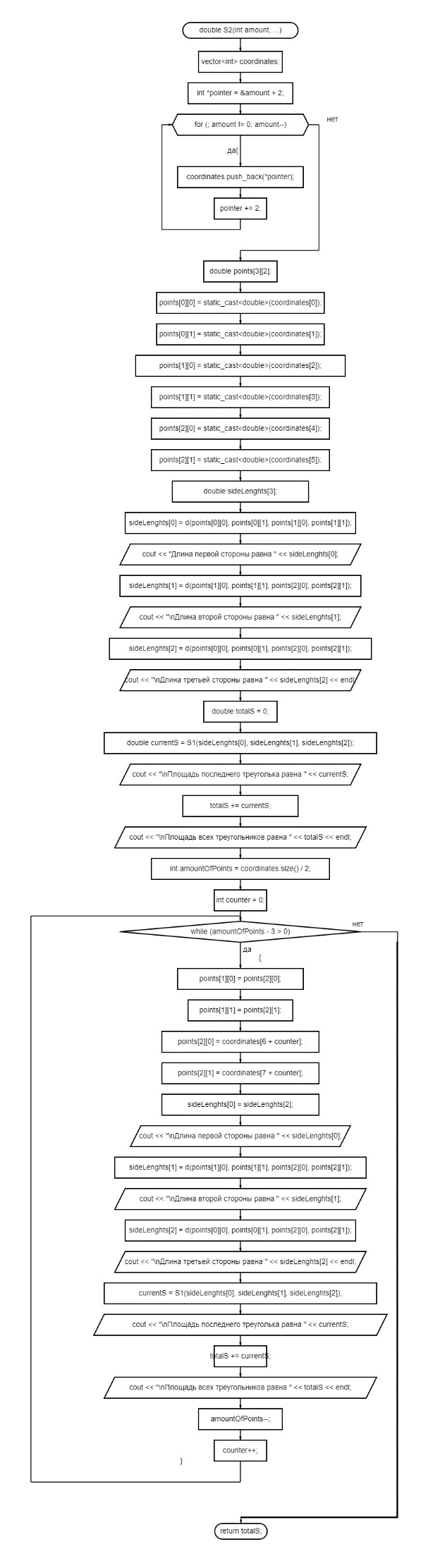
1. Функция `d()` находит расстояние между двумя точками по их координатам с помощью формулы расстояния между двумя точками на плоскости.
2. Функция `S1()` вычисляет площадь треугольника с использованием формулы Герона.
3. Функция `S2()` рассчитывает площадь выпуклого многоугольника, заданного координатами его вершин, с помощью метода разбиения на треугольники. Она принимает переменное количество параметров, где каждые два параметра представляют собой координаты точки. Функция формирует треугольники из заданных точек и суммирует их площади.

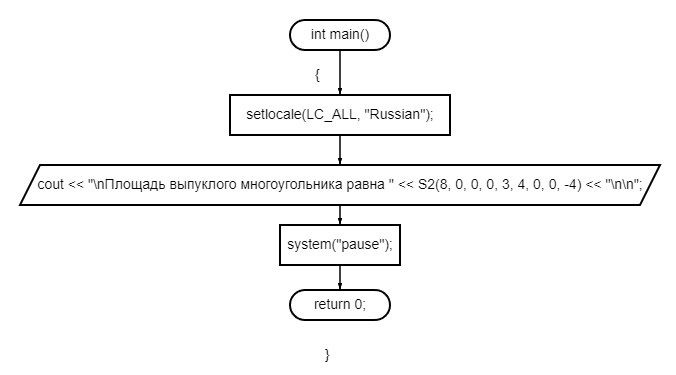
**Моделирование:**



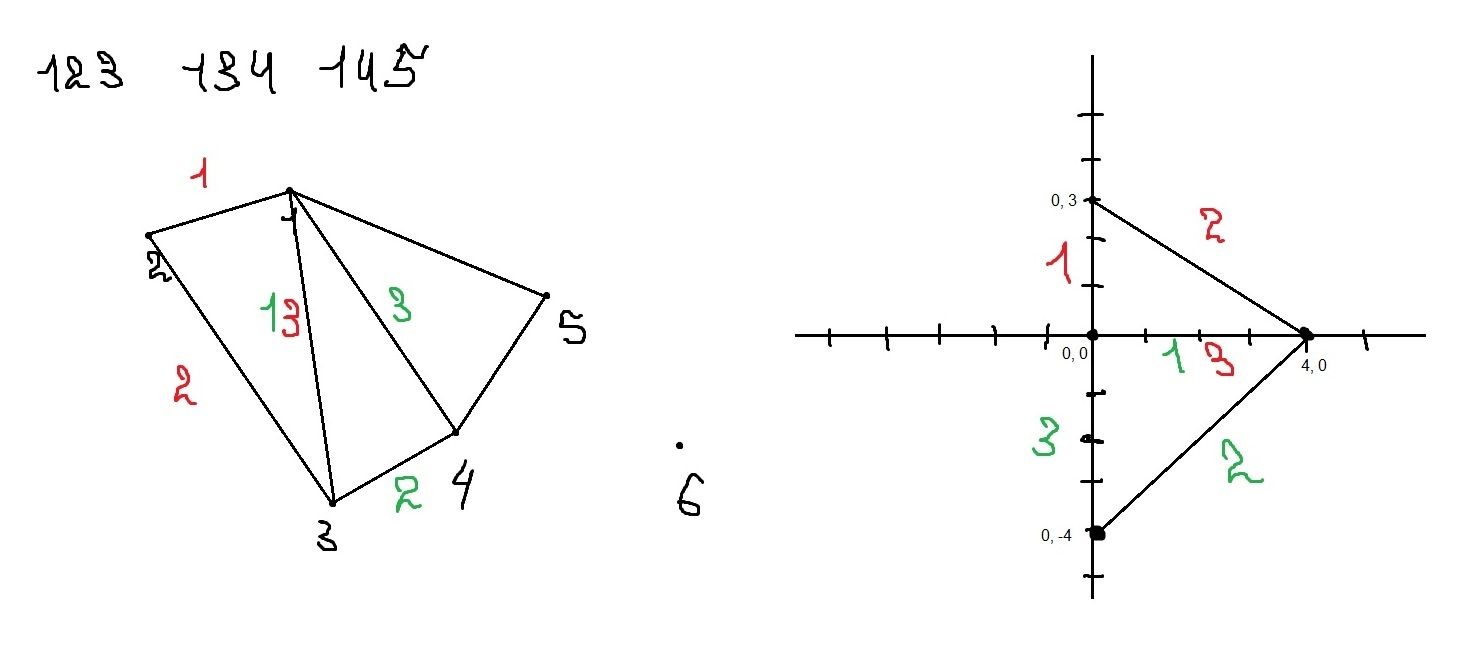


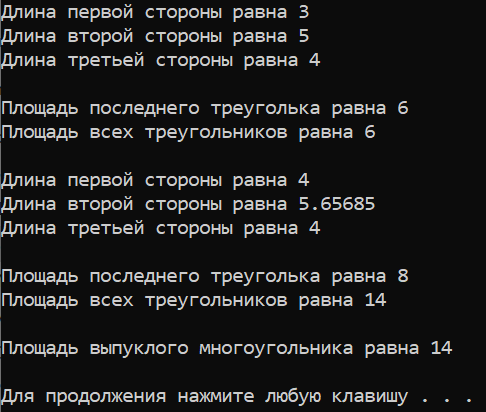






**Результаты работы программы:**





**Код программы:**

#include <iostream>

#include <cmath>

#include <vector>

using namespace std;

// Нахождение длины стороны по формуле расстояния между двумя точками

double d(double x1, double y1, double x2, double y2)

{

return sqrt(pow((x1 - x2), 2) + pow((y1 - y2), 2));

}

// Нахождение площади треугольника по формуле Герона

double S1(double a, double b, double c)

{

double p = (a + b + c) / 2;

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

}

// Нахождение площади выпуклого многоугольника

double S2(int amount, ...)

{

// Передача параметов в массив-вектор

vector<int> coordinates;

int \*pointer = &amount + 2; // Указатель на первый неявный параметр функции

for (; amount != 0; amount--)

{

coordinates.push\_back(\*pointer);

pointer += 2;

}

// Работа с первыми тремя точками

double points[3][2];

points[0][0] = static\_cast<double>(coordinates[0]);

points[0][1] = static\_cast<double>(coordinates[1]);

points[1][0] = static\_cast<double>(coordinates[2]);

points[1][1] = static\_cast<double>(coordinates[3]);

points[2][0] = static\_cast<double>(coordinates[4]);

points[2][1] = static\_cast<double>(coordinates[5]);

double sideLenghts[3];

sideLenghts[0] = d(points[0][0], points[0][1], points[1][0], points[1][1]);

cout << "Длина первой стороны равна " << sideLenghts[0];

sideLenghts[1] = d(points[1][0], points[1][1], points[2][0], points[2][1]);

cout << "\nДлина второй стороны равна " << sideLenghts[1];

sideLenghts[2] = d(points[0][0], points[0][1], points[2][0], points[2][1]);

cout << "\nДлина третьей стороны равна " << sideLenghts[2] << endl;

double totalS = 0;

double currentS = S1(sideLenghts[0], sideLenghts[1], sideLenghts[2]);

cout << "\nПлощадь последнего треуголька равна " << currentS;

totalS += currentS;

cout << "\nПлощадь всех треугольников равна " << totalS << endl;

// Универсальная часть программы

int amountOfPoints = coordinates.size() / 2;

int counter = 0;

while (amountOfPoints - 3 > 0)

{

points[1][0] = points[2][0];

points[1][1] = points[2][1];

points[2][0] = coordinates[6 + counter];

points[2][1] = coordinates[7 + counter];

sideLenghts[0] = sideLenghts[2];

cout << "\nДлина первой стороны равна " << sideLenghts[0];

sideLenghts[1] = d(points[1][0], points[1][1], points[2][0], points[2][1]);

cout << "\nДлина второй стороны равна " << sideLenghts[1];

sideLenghts[2] = d(points[0][0], points[0][1], points[2][0], points[2][1]);

cout << "\nДлина третьей стороны равна " << sideLenghts[2] << endl;

currentS = S1(sideLenghts[0], sideLenghts[1], sideLenghts[2]);

cout << "\nПлощадь последнего треуголька равна " << currentS;

totalS += currentS;

cout << "\nПлощадь всех треугольников равна " << totalS << endl;

amountOfPoints--;

counter++;

}

return totalS;

}

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "\nПлощадь выпуклого многоугольника равна " << S2(8, 0, 0, 0, 3, 4, 0, 0, -4) << "\n\n"; // 8 параметров: координаты точек (0; 0), (0; 3), (4; 0), (0; -4)

system("pause");

return 0;

}

**Ссылка на Git:** <https://github.com/Exateym/Study>