«Отчет по ПЗ 1.1. Динамические массивы (одномерные)» Никита Преображенский \*\*\*. Вариант 30.

1. Задание: каждая пара (*Xk, Yk*) представляет длины сторон треугольника, а их общее основание имеет длину *B*. Для каждой из *n* пар *(Xk, Yk)* про­ве­рить возможность построения треугольника. Найти сумму площа­дей построенных треугольников.
2. Постановка задачи.

Входные данные + ограничения:

А) Переменная n. Тип int. Размер массивов (кол-во пар сторон треугольников).

Исключения: буква, строка, вещественное, меньше или равно 0.

Б) Переменная B. Тип double. Длина стороны основания.

Исключения: буква, строка, отрицательные числа, числа равные 0.

В) Переменные X, Y. Тип \*double[]. Динамические массивы длин сторон треугольников.

Исключения: буква, строка, отрицательные числа, числа равные 0.

Выходные данные + ограничения:

А) Переменная sum. Тип double. Сумма площадей построенных треугольников.

Исключения: ограничения значения заданными пределами операционной системы устройства для типа double.

Промежуточные данные + ограничения:

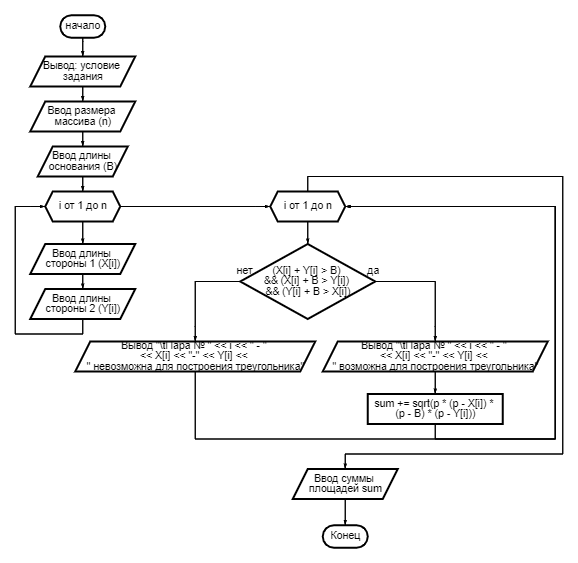
А) Переменная p. Тип double. Полупериметр созданного треугольника.

Исключения: ограничения значения заданными пределами операционной системы устройства для типа double.

Среда разработки: Visual Studio 2022.

2. Структура программы.

2.1 Блок-схема алгоритма



3. Тесты.

|  |  |  |
| --- | --- | --- |
| Входные данные | Результат программы | Смысл |
| n = a | Неверное значение длины массивов! | Проверка на букву |
| n = 3a | Неверное значение длины массивов! | Проверка на строку |
| n = 3,2 | Неверное значение длины массивов! | Проверка на целое число |
| n = 10000000000000000 | Неверное значение длины массивов! | Выделение памяти массивам |
| n = 0 | Неверное значение длины массивов! | Валидность размера массива |
| B = a | Incorrect input! | Проверка на букву |
| B = 2a | Incorrect input! | Проверка на строку |
| X[i] = d | Неверное значение для элемента массива! | Проверка на букву |
| X[i] = 3a | Неверное значение для элемента массива! | Проверка на число c буквой |
| X[i] = dsadas | Неверное значение для элемента массива! | Проверка на строку |
| Y[i] = d | Неверное значение для элемента массива! | Проверка на букву |
| Y[i] = 3d | Неверное значение для элемента массива! | Проверка на число c буквой |
| Y[i] = dsaklsa | Неверное значение для элемента массива! | Проверка на строку |
| n = 2  B = 6  X[i] = 3  Y[i] = 5  X[i] = 1  Y[i] = 2 | Пара № 0 - 3-5 возможна для построения треугольника  Пара № 1 - 1-2 невозможна для построения треугольника  Сумма площадей построенных треугольников равна: 7.48331 | Проверка на корректность работы в случае, когда имеется возможный и невозможный треугольник. |
| n = 1  B = 6  X[i] = 1  Y[i] = 2 | Пара № 0 - 1-2 невозможна для построения треугольника  Нет подходящих треугольнико! | Проверка корректности работы программы при отсутствии подходящих треугольников. |

4. Код:

#include <iostream> // Подключение стандартной библиотеки

#include <cmath> // Подключение стандартной математической библиотеки

void InputInt(int\*); // Прототип функции ввода размера массива

void InputDouble(double\*); // Прототип функции ввода элементов массива

using namespace std; // Использование стандартного пространства имён

int main(void)

{

setlocale(LC\_ALL, "RUS"); // Установка для консоли русского языка

cout << "\t\tВариант № 30: Каждая пара (Xk, Yk) представляет длины сторон треугольника, а их общее основание имеет длину B.";

cout << "\n\t\tДля каждой из n пар(Xk, Yk) проверить возможность построения треугольника.";

cout << "\n\t\tНайти сумму площадей построенных треугольников." << endl;

int n = 0; // Размер массивов

double B = 0; // Длина основания треугольников

cout << endl << "\tВведите размер массивов:\n";

InputInt(&n); // Вызов функции ввода размера массива

cout << endl <<"\tВведите длину основания B:\n";

InputDouble(&B); // Вызов функции ввода элементов массива

double\* X = new double[n]; // Создание массива X на n элементов

double\* Y = new double[n]; // Создание массива Y на n элементов

cout << endl << "\tВведите " << n << " пар-(ы) точек:\n";

for (int i = 0; i < n; ++i) // Заполнение массивов

{

cout << endl << "\tВведите № " << i << " X координату:\n";

InputDouble(&X[i]);

cout << endl << "\tВведите № " << i << " Y координату:\n";

InputDouble(&Y[i]);

}

double sum = 0; // Переменная суммы площадей созданных треугольников

double p = 0; // Временная переменная полупериметр

for (int i = 0; i < n; ++i) // Обработка элементов

{

if ((X[i] + Y[i] > B) && (X[i] + B > Y[i]) && (Y[i] + B > X[i])) // Проверка условий на возможность создания треугольника

{

cout << endl << "\tПара № " << i << " - " << X[i] << "-" << Y[i] << " возможна для построения треугольника" << endl;

p = (X[i] + Y[i] + B)/2; // Расчёт полупериметра треугольника

sum += sqrt(p \* (p - X[i]) \* (p - B) \* (p - Y[i])); // Сложение к сумме

}

else

{

cout << endl << "\tПара № " << i << " - " << X[i] << "-" << Y[i] << " невозможна для построения треугольника" << endl;

}

}

if (sum != 0)

{

cout << endl << "\tСумма площадей построенных треугольников равна: " << sum << endl; // Вывод суммы площадей подходящих треугольников

}

else

{

cout << "Нет подходящих треугольников!" << endl;

}

return 0;

}

void InputInt(int\* tmp) // Описание функций ввода размера массива

{

while (!(cin >> \*tmp) || (cin.peek() != '\n') || (\*tmp <= 0))

{

cin.clear();

while (cin.get() != '\n');

cout << "Incorrect input!" << endl;

}

}

void InputDouble(double\* tmp) // Описание функции ввода элементов массивов

{

while (!(cin >> \*tmp) || (cin.peek() != '\n') || (\*tmp <= 0))

{

cin.clear();

while (cin.get() != '\n');

cout << "Incorrect input!" << endl;

}

}