**НИУ «МЭИ»**

**Лабораторная работа № 7**

Нисходящее проектирование: программы с использованием

подпрограмм и модулей

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил: \_\_Кобзев Н.И.\_\_

Группа: \_\_\_Э-07-20\_\_\_

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ф.И.О./

Москва, 2020 г.

1. **Формулировка исходной задачи:**

Заданы, распределенные на N пар, 2N треугольников длинами их сторон A1, B1, C1 (1-й треугольник каждой пары) и A2, B2, C2 (2-й треугольник), причем A1 < B1 < C1 и A2 < B2 < C2. Определить число случаев подобия треугольников, входящих в пару, а также подсчитать суммарные площадь и периметр каждой пары подобных треугольников.

1. **Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Имя, смысл | Тип | Структура | Диапазон |
| Входные данные | n – количество пар треугольников | Целый | Простая переменная | 1 ≤ n ≤ 10 |
| triangles1 – массив первых в паре треугольников | Целый | Одномерный массив | [1, 100] |
| triangles2 – массив вторых в паре треугольников | Целый | Одномерный массив | [1, 100] |
| Промежуточные данные | i – счетчик цикла | Целый | Простая переменная | 0 ≤ i ≤ n |
| j – счетчик цикла | Целый | Простая переменная | 0 ≤ j ≤ 3 |
| g – счетчик цикла | Целый | Простая переменная | 0 ≤ g ≤ n |
| Выходные данные | count – количество подобных пар | Целый | Простая переменная | 0 ≤ count ≤ 10 |
| square – массив площадей | Дробный | Одномерный массив | [2, 20000] |
| perimeter – массив периметров | Дробный | Одномерный массив | [6, 600] |

1. **Блок-схема:**

**+**

-

count != 0

-

square.append(finding\_the\_area(triangles1[i]) + (finding\_the area(triangles2[i])

perimeter.append(finding\_the\_perimeter(triangles1[i]) + (finding\_the perimeter(triangles2[i])

count = count + 1

-

-

**+**

**+**

**+**

not similarity\_check(triangles1[i], triangles2[i]

not is\_there\_the\_triangle(triangles1[i], triangles2[i]

for i in range(n)

count = 0; square = []; perimeter = []

Ввод n, triangles1, triangles2

Вывод count

for elem in square

**+**

Вывод elem

-

for elem in perimeter

**+**

-

Вывод elem

Вывод сообщения, что подобных треугольников – нет

1. **Тесты:**

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Смысл теста |
| 3  2 3 4 1 2 3  2 3 4 4 6 8  1 1 1 3 3 3 | Треугольника со сторонами (1, 2, 3) не существует  Количество пар подобных треугольников равно: 2  Суммарные площади каждой пары подобных треугольников равны:  14.523687548277813  4.330127018922194  Суммарные периметры каждой пары подобных треугольников равны:  27  12 | Проверка условия существования треугольника и выполнения условия |

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Смысл теста |
| 2  2 3 4 1 2 3  2 3 4 3 6 8 | Треугольника со сторонами (1, 2, 3) не существует  Подобные треугольники не были найдены | Проверка условия существования треугольника и невыполнения условия |

1. **Код программы:**

import math  
import sys  
  
  
def is\_there\_the\_triangle(triangle1, triangle2):  
 a1 = triangle1[0]  
 b1 = triangle1[1]  
 c1 = triangle1[2]  
 a2 = triangle2[0]  
 b2 = triangle2[1]  
 c2 = triangle2[2]  
 if a1 <= 0 or b1 <= 0 or c1 <= 0 or a2 <= 0 or b2 <= 0 or c2 <= 0:  
 print("Треугольника с отрицательными сторонами не бывает)")  
 return False  
 if not (a1 + b1 > c1 and a1 + c1 > b1 and b1 + c1 > a1):  
 print("Треугольника со сторонами ({}, {}, {}) не существует".format(a1, b1, c1))  
 return False  
 if not (a2 + b2 > c2 and a2 + c2 > b2 and b2 + c2 > a2):  
 print("Треугольника со сторонами ({}, {}, {}) не существует".format(a2, b2, c2))  
 return False  
 return True  
  
  
def similarity\_check(triangle1, triangle2):  
 if triangle1[0] / triangle2[0] == triangle1[1] / triangle2[1] == triangle1[2] / triangle2[2]:  
 return True  
 else:  
 return False  
  
  
def finding\_the\_area(triangle):  
 p = (triangle[0] + triangle[1] + triangle[2]) / 2  
 s = math.sqrt(p \* (p - triangle[0]) \* (p - triangle[1]) \* (p - triangle[2]))  
 return s  
  
  
def finding\_the\_perimeter(triangle):  
 return triangle[0] + triangle[1] + triangle[2]

name\_file = sys.argv[1]  
  
file = open(name\_file, 'r')  
  
n = int(file.readline())  
  
triangles1 = [[] for i in range(n)]  
triangles2 = [[] for g in range(n)]  
  
for i in range(n):  
 \_ = file.readline().split()  
 for j in range(3):  
 triangles1[i].append(int(\_[j]))  
 triangles2[i].append(int(\_[3 + j]))  
  
count = 0  
square = []  
perimeter = []  
  
for i in range(n):  
 if not is\_there\_the\_triangle(triangles1[i], triangles2[i]):  
 continue  
  
 if not similarity\_check(triangles1[i], triangles2[i]):  
 continue  
 else:  
 count += 1  
  
 square.append(finding\_the\_area(triangles1[i]) + finding\_the\_area(triangles2[i]))  
  
 perimeter.append(finding\_the\_perimeter(triangles1[i]) + finding\_the\_perimeter(triangles2[i]))  
  
if count != 0:  
 print("Количество пар подобных треугольников равно: {}".format(count))  
 print("Суммарные площади каждой пары подобных треугольников равны:")  
 for elem in square:  
 print("{}".format(elem))  
 print("Суммарные периметры каждой пары подобных треугольников равны:")  
 for elem in perimeter:  
 print("{}".format(elem))  
else:  
 print("Подобные треугольники не были найдены")  
print()