**НИУ «МЭИ»**

**Лабораторная работа № 8**

Работа с несколькими массивами

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. **УПЗ:** Задаются N пар треугольников, каждый треугольник из пары записывается в свою двумерную матрицу A[N][3], где 3 – это количество сторон треугольника. Затем мы проверяем подобны ли треугольники. Если да, то ищем периметр и площадь и записываем их в выходные массивы perimeter и square соответственно.
2. **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 | i = n |
| j – счетчик цикла | Целый | Простая переменная | j = 3 |
| a1 – сторона треугольника 1 | Целый | Простая переменная | [1, 100] |
| b1 – сторона треугольника 1 | Целый | Простая переменная | [1, 100] |
| c1 – сторона треугольника 1 | Целый | Простая переменная | [1, 100] |
| a2 – сторона треугольника 2 | Целый | Простая переменная | [1, 100] |
| b2 – сторона треугольника 2 | Целый | Простая переменная | [1, 100] |
| c2 – сторона треугольника 2 | Целый | Простая переменная | [1, 100] |
| flg – переменная для возврата значения из функции |  | Простая переменная | Истина или Ложь |
| p – полумериметр | Вещественный | Простая переменная | [1, 150] |
| s – площадь | Вещественный | Простая переменная | [1; 4500] |
| elem – итератор по массивам | Вещественный | Простая переменная | [1, 20000] |
| Выходные данные | count – количество подобных пар | Целый | Простая переменная | 0 ≤ count ≤ 10 |
| square – массив площадей | Вещественный | Одномерный массив | [2, 20000] |
| perimeter – массив периметров | Вещественный | Одномерный массив | [6, 600] |

1. **Блок-схема:**

Ввод n, triangles1, triangles2

**+**

-

count != 0

-

count = count + 1

**+**

i от 0 до n

Вх.: triangles1[i], triangles2[i]

Вх.: triangles1[i], triangles2[i]

Вых.: flg

-

flg

Проверка на подобие треугольников

-

flg

Вых.: flg

Проверка на существование треугольника

perimeter[i] = нахождение периметра первого треугольника + нахождение периметра второго треугольника

square[i] = нахождение площади первого треугольника + нахождение площади второго треугольника

**+**

**+**

Вывод n, triangles1, triangles2

Вых.: p

Вх.: triangles(1/2)[i]

Вых.: s

Вх.: triangles(1/2)[i]

count = 0; square = [0] \* n; perimeter = [0] \* n

Вывод elem

elem != 0

-

Вывод elem

elem != 0

**+**

elem в perimeter

Вывод count

elem в square

-

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): # Проверка существования треугольника |
| check = True |
| 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)") |
| check = False |
| if not (a1 + b1 > c1 and a1 + c1 > b1 and b1 + c1 > a1): |
| print("Треугольника со сторонами ({}, {}, {}) не существует".format(a1, b1, c1)) |
| check = False |
| if not (a2 + b2 > c2 and a2 + c2 > b2 and b2 + c2 > a2): |
| print("Треугольника со сторонами ({}, {}, {}) не существует".format(a2, b2, c2)) |
| check = False |
| return check |
|  |
|  |
| def similarity\_check(triangle1, triangle2): # Проверка подобия |
| if triangle1[0] / triangle2[0] == triangle1[1] / triangle2[1] == triangle1[2] / triangle2[2]: |
| check = True |
| else: |
| check = False |
| return check |
|  |
|  |
| def finding\_the\_area(triangle): # нахождение площади |
| 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): # нахождение периметра |
| return triangle[0] + triangle[1] + triangle[2] |
|  |
|  |
| name\_file = sys.argv[1] # чтение командной строки |
|  |
| file = open(name\_file, 'r') # открытие файла |
|  |
| n = int(file.readline()) # считывание количества треугольников |
|  |
| # инициализация массивов |
| triangles1 = [0] \* n |
| triangles2 = [0] \* n |
|  |
| for i in range(n): |
| triangles1[i] = [0] \* 3 |
| triangles2[i] = [0] \* 3 |
|  |
| for i in range(n): # считывание массивов |
| line = file.readline().split() |
| for j in range(3): |
| triangles1[i][j] = int(line[j]) |
| triangles2[i][j] = int(line[3 + j]) |
|  |
| # инициализация переменных |
| count = 0 |
| square = [0] \* n |
| perimeter = [0] \* n |
|  |
| for i in range(n): |
| if is\_there\_the\_triangle(triangles1[i], triangles2[i]): # существуют ли треугольники |
| if similarity\_check(triangles1[i], triangles2[i]): # подобны ли треугольники |
| # подсчет количества подобных треуг и нахождения площади и периметра |
| count += 1 |
| square[i] = (finding\_the\_area(triangles1[i]) + finding\_the\_area(triangles2[i])) |
| perimeter[i] = (finding\_the\_perimeter(triangles1[i]) + finding\_the\_perimeter(triangles2[i])) |
|  |
| if count != 0: # проверка на количество подобных треугольников |
| print("Количество пар подобных треугольников равно: {}".format(count)) |
| print("Суммарные площади каждой пары подобных треугольников равны:") |
| for elem in square: # вывод площадей |
| if elem != 0: |
| print("{}".format(elem)) |
| print("Суммарные периметры каждой пары подобных треугольников равны:") |
| for elem in perimeter: # вывод периметров |
| if elem != 0: |
| print("{}".format(elem)) |
| else: |
| print("Подобные треугольники не были найдены") |
| print() |