|  |  |  |
| --- | --- | --- |
| A | B | C |
| 1 | 2 | 7 |
| 1 | 7 | 2 |
| 7 | 2 | 1 |
| 2 | 2 | 2 |
| 2 | 2 | 3 |
| 2 | 3 | 4 |
| -1 | 3 | 4 |
| 3 | -1 | 4 |
| 3 | 4 | -1 |
| 2 | 3 | 4 |
| 2 | 3 | 2 |
| 2 | 2 | 4 |
| 4 | 2 | 2 |
| 2 | 4 | 2 |
| 3 | 2 | 2 |

import org.junit.\*;

import java.util.Random;

import static org.junit.Assert.assertEquals;

public class tritypTest {

trityp tri=new trityp();

// Triang = 1 if triangle is scalene

// Triang = 2 if triangle is isosceles

// Triang = 3 if triangle is equilaterala

// Triang = 4 if not a triangle

@Test

public void testTriang\_4(){

assertEquals(4,tri.Triang(1,2,7));

assertEquals(4,tri.Triang(1,7,2));

assertEquals(4,tri.Triang(7,2,1));

assertEquals(3,tri.Triang(2,2,2));

assertEquals(2,tri.Triang(2,2,3));

assertEquals(1,tri.Triang(2,3,4));

assertEquals(4,tri.Triang(-1,3,4));

assertEquals(4,tri.Triang(3,-1,4));

assertEquals(4,tri.Triang(3,4,-1));

assertEquals(1,tri.Triang(2,3,4));

assertEquals(2,tri.Triang(2,3,2));

assertEquals(4,tri.Triang(2,2,4));

assertEquals(4,tri.Triang(4,2,2));

assertEquals(4,tri.Triang(2,4,2));

assertEquals(4,tri.Triang(3,2,2));

}

}