#include <iostream>

using namespace std;

class CTriangle {

public:

CTriangle() {

m\_first\_side = 0;

m\_second\_side = 0;

m\_third\_side = 0;

m\_square = 0;

m\_radius = 0;

}

CTriangle(double square) {

m\_square = square;

m\_first\_side = 0;

m\_second\_side = 0;

m\_third\_side = 0;

m\_radius = 0;

}

CTriangle(double first\_side, double second\_side, double third\_side) {

m\_first\_side = first\_side;

m\_second\_side = second\_side;

m\_third\_side = third\_side;

m\_radius = 0;

double p = (first\_side + second\_side + third\_side) / 2;

m\_square = sqrt(p \* (p - first\_side) \* (p - second\_side) \* (p - third\_side));

}

CTriangle(const CTriangle& other) {

this->m\_first\_side = other.m\_first\_side;

this->m\_second\_side = other.m\_second\_side;

this->m\_third\_side = other.m\_third\_side;

this->m\_square = other.m\_square;

this->m\_radius = other.m\_radius;

}

double FindRadiusOfInnerCircle(double closer\_to\_base, double closer\_to\_same, double square) {

m\_square = square;

double x = pow(pow(m\_square, 2) / (pow(closer\_to\_base, 2) \* (pow(closer\_to\_base + closer\_to\_same, 2) - pow(closer\_to\_base, 2))), 0.25);

m\_first\_side = m\_second\_side = x \* (closer\_to\_base + closer\_to\_same);

m\_third\_side = 2 \* closer\_to\_base \* x;

double p = (m\_first\_side + m\_second\_side + m\_third\_side) / 2;

double radius = square / p;

m\_radius = radius;

return radius;

}

bool operator<(const CTriangle& other) {

if (this->m\_radius < other.m\_radius)

return true;

return false;

}

bool operator<=(const CTriangle& other) {

if (this->m\_radius <= other.m\_radius)

return true;

return false;

}

bool operator>(const CTriangle& other) {

if (this->m\_radius > other.m\_radius)

return true;

return false;

}

bool operator>=(const CTriangle& other) {

if (this->m\_radius >= other.m\_radius)

return true;

return false;

}

bool operator==(const CTriangle& other) {

if (this->m\_radius == other.m\_radius)

return true;

return false;

}

bool operator!=(const CTriangle& other) {

if (this->m\_radius == other.m\_radius)

return true;

return false;

}

double GetRadius() { return m\_radius; }

double GetSquare() { return m\_square; }

private:

//Тут немає що виділяти динамічно)

double m\_first\_side;

double m\_second\_side;

double m\_third\_side;

double m\_square;

double m\_radius;

};

int main()

{

CTriangle tr\_1, tr\_2, tr\_3;

tr\_1.FindRadiusOfInnerCircle(12, 25, 1680);

tr\_2.FindRadiusOfInnerCircle(17, 22, 1680);

tr\_3.FindRadiusOfInnerCircle(22, 17, 1680);

CTriangle tr\_4(tr\_3);

cout << "Relation 12/25";

if (tr\_1 > tr\_2)

cout << " > ";

else

cout << " < ";

cout << "17/22\n\n";

cout << "Relation 12/25 ";

if (tr\_1 > tr\_4)

cout << " > ";

else

cout << " < ";

cout << "22/17\n\n";

}