#include<iostream>

#include<cmath>

using namespace std;

class triang {

private:

double a;

double b;

double c;

public:

triang() {

a = 0;

b = 0;

c = 0;

}

triang(double sA, double sB, double sC) {

//if (sA+sB<sC || sA+sC<sB || sB+sC<sA) {

// cout << "Warning: uncorrect sides" << endl;

a = sA;

b = sB;

c = sC;

}

///////////

void setA(double value) {

a = value;

}

void setB(double value) {

b = value;

}

void setC(double value) {

c = value;

}

double getA() {

return a;

}

double getB() {

return b;

}

double getC() {

return c;

}

///////////

void showContent() {

cout << "sides: " << a << ", " << b << ", " << c << endl

<< "square: " << square() << endl

<< "perimeter: " << perimeter() << endl << endl;

}

void changeContent() {

cout << "side a: ";

cin >> a;

cout << "side b: ";

cin >> b;

cout << "side c: ";

cin >> c;

}

void changeContent(double sA, double sB, double sC) {

a = sA;

b = sB;

c = sC;

}

bool checkSides() {

if(a+b<c && a+c<b && b+c<a)

return true;

return false;

}

double square() {

double p = (a+b+c)/2;

return sqrt(p\*(p-a)\*(p-b)\*(p-c));

}

double perimeter() {

return a+b+c;

}

};

void sort(triang arr[], int n) {

triang temp;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j].perimeter() > arr[j + 1].perimeter()) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

triang maxValue(triang arr[], int n) {

triang max = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i].square() > max.square())

max = arr[i];

}

return max;

}

void main() {

triang tr1(10, 17, 21);

triang tr2(7,11,12);

triang tr3(5, 4, 3);

const int n = 3;

triang arr[n];

arr[0] = tr1;

arr[1] = tr2;

arr[2] = tr3;

///////////

sort(arr, n);

for (int i = 0; i < n; i++) {

arr[i].showContent();

}

///////////

maxValue(arr, n).showContent();

///////////

triang tr4;

tr4.changeContent();

tr4.showContent();

}