pb Triángulo

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 4MB |
| Límite de tiempo (caso) | 0.25s | Límite de tiempo (total) | 60s |

**Descripción**

Un triángulo es formado por 3 segmentos (con valor entero) de recta unidos por sus extremos, sin embargo no todas las combinaciones de 3 segmentos pueden formar un triángulo.

**Problema**

Escriba un programa que dados 3 números que representan 3 segmentos de rectas, diga si pueden formar un triángulo rectángulo.

**Entrada**

En la primera línea 3 números “a”, “b” y “c” representando las longitudes de 3 segmentos de recta en un rango de entre 1 y 32,000.

**Salida**

Una sola línea en donde si es posible generar un triángulo rectángulo debes imprimir la hipotenusa, el cateto mayor y el cateto menor y si no es posible formar un triángulo rectángulo imprime la palabra “imposible”.

**Consideraciones**

Tu programa se evaluará con varios casos de prueba, para cada caso de prueba (entrada)

**Ejemplos**

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| 1 1 1 | imposible | Caso #1 |
| 3 4 5 | 5 4 3 | Caso #2 |

**Consideraciones**

Tu programa se ejecutara con varios casos de prueba.

**Tips**

El teorema de Pitágoras la hipotenusa es igual a la raíz cuadrada de la suma de los cuadrados de sus catetos.

¿Es necesario hacer el cálculo de la raíz cuadrada?

*Fuente: Curso de Profesores*

Problema subido por: [CursoProfesores](https://omegaup.com/profile/CursoProfesores/)

<https://omegaup.com/arena/problem/pb-Triangulo#problems>

#include <iostream>

#include <stdio.h>

#include <vector>

#include <algorithm>

#include <conio.h>

using namespace std;

int main() {

int a,b,c;

scanf("%d %d %d", &a, &b, &c);

std::vector<int> lados ;

lados.push\_back(a);

lados.push\_back(b);

lados.push\_back(c);

std::sort(lados.begin(), lados.end());

if(lados[0] \* lados[0] + lados[1] \* lados[1] == lados[2] \* lados[2]){

std::reverse(lados.begin(), lados.end());

for(int i =0; i < lados.size(); i++) {

printf("%d ", lados[i]);

}

}else {

printf("imposible");

}

getch();

return 0;

}