Problema 5

Descripción

¿Recuerdas cuando en la escuela calculabas áreas y perímetros de figuras geométricas? Calcular estos valores sabiendo los lados de la figura seguramente te resultaba bastante trivial. ¿Qué tan complejo sería hacer lo contrario? En este problema nos enfocaremos en el proceso inverso: Es decir, dados el perímetro y área de un triángulo, tu tarea será encontrar todos los triángulos con lados enteros positivos que tengan dicho perímetro y área.

Entrada

2 números (pueden contener decimales) separados por una línea. El primer número representa el perímetro y el segundo el área.

Salida

En caso de que existan triángulos que cumplan las condiciones del perímetro y área, cada línea debe constar de 3 enteros positivos (los lados del triángulo) separados por un espacio y ordenados ascendentemente. En caso de existir múltiples triángulos, las líneas de la salida deben estar ordenadas lexicográficamente entre sí.

En caso de no existir ningún triángulo que cumpla, imprimir la cadena "no hay".

Ejemplo

entrada	salida
1 1	no hay
70 210	17 25 28 20 21 29

Notas

- Recuerda que un triángulo es válido solamente si sus lados cumplen con las desigualdades triangulares: Para valores de sus lados a, b, c se debe cumplir que $c < a + b$, $a < b + c$, $b < a + c$.
- Los valores del perímetro y área dados serán igual o menores a 10000.
- 2 números de tipo float se consideran iguales si difieren en menos de 0.00001.

Solución:
Como un primer intento de solución a fuerza bruta, podemos hacer una búsqueda con un ciclo triple para los lados, respetando las desigualdades triangulares. Luego calculamos el perímetro y el área y verificamos si son iguales al perímetro y área dada por la entrada del problema. Para calcular el área en base a los 3 lados de un triángulo, podemos utilizar la fórmula de Herón. Para un triángulo con lados l_1, l_2, l_3 su área A está dada por la fórmula:

$$A = \sqrt{s(s - l_1)(s - l_2)(s - l_3)}$$

donde $s = \frac{l_1+l_2+l_3}{2}$ es el semiperímetro de la figura.

Esta solución no es del todo eficiente, ya que tiene un ciclo de más. Ya que tenemos el perímetro dado, bastaría hacer una búsqueda para dos de los lados del triángulo con un doble ciclo. El tercer lado lo podemos deducir restándole ambos lados considerados al perímetro. Incluso se puede acotar la búsqueda usando la desigualdad triangular para quitar casos imposibles, pero esta última optimización no afecta a la complejidad de tiempo de la solución.

Aquí mostramos la solución simple con ciclo triple, pues es suficiente para resolver los casos del problema. Supongamos sin pérdida de generalidad que $l_1 \geq l_2 \geq l_3$. La desigualdad triangular implica que $2l_1 < l_2 + l_3 + l_1$ o que $l_1 < \frac{p}{2}$ donde p es el perímetro del triángulo. También implica que $l_2 + l_3 > l_1$, de donde $l_3 \geq l_1 - l_2 + 1$. Los límites de l_1, l_2 y l_3 están entonces dados por:

$$
\begin{aligned}
1 \leq l_1 &< \frac{p}{2} \\
1 \leq l_2 &\leq l_1 \\
l1 - l2 + 1 \leq l_3 &\leq l_2.
\end{aligned}
$$

Con estos límites, podemos armar los tres ciclos correctamente:

```
#include <iostream>
#include <cmath>
using namespace std;

int main ()
{
    double p;
    double a;
    cin >> p;
    cin >> a;
    bool hay = 0;
    double sp = p / 2;

    for (int l1 = 1; l1 <= p/2; l1++)
    {
        for (int l2 = 1; l2 <= l1; l2++)
        {
            for (int l3 = l1-l2+1; l3 <= l2; l3++)
            {
                if (l1 + l2 + l3 == p
                    && abs(sqrt(sp * (sp-l1) * (sp-l2) * (sp-l3)) - a) < 0.00001)
                {
                    hay = true;
                    cout << " " << l2 << " " << l1 << endl;
                }
            }
        }
    }

    if (!hay)
    {
        cout << "no hay";
    }
}
```