In a right triangle, the square of the length of the hypotenuse is equal to the sum of the squares of the lengths of the other two sides. For example, a triangle with sides of lengths 4, 5, and 3 units is a right triangle because 5^2 (25) = 3^2 (9) + 4^2 (16).

Write a program that prompts for the **_integer_** lengths of the three sides of a triangle and prints an appropriate message saying whether the triangle is a right triangle or not. You may not assume the sides are entered in any particular order, but you may assume the sides would make a triangle (the sum of the lengths of two sides will be greater than the length of the third side). You may assume all sides will have integer lengths greater than 0.

Test with 3, 4, 5; 5, 4, 3; 4, 5, 3; 6, 10, 8; 4, 4, 6; and several other sets of lengths of your own choice, both forming a right triangle and not forming a right triangle.

Use three separate if() statements for the "is-it-a-right-triangle" tests, one for each possible hypotenuse. **_Remember_** if you don't find a right triangle for a fourth if() statement printing a message saying the sides don't form a right triangle. How do you remember something in a program?

You may not use if-else() for this program. For your fourth if() test, you _MAY NOT_ just test "is it not a right triangle" by using the sides again. You may not use compound tests (e.g., using `&&` or `||`).

You may have only one `return 0;` statement, at the bottom of main(). Do not return from <u>any</u> of the if() statements. To square a number, just multiply it by itself: please do not use pow()! Do not use floating-point types for anything in this assignment.

For example, a test run might look like this:
```
Enter side a : 4
Enter side b : 3
Enter side c : 5
The sides form a right triangle.
```

Or, it might look like this:
```
Enter side a : 4
Enter side b : 4
Enter side c : 7
The sides do not form a right triangle.
```