

# Euler 0009

## The Problem:

A pythagorean triple is such that integers a, b, c can be  $a^2 + b^2 = c^2$ .

There exists a pythagorean triple that sum total of  $a + b + c = 1000$ . Find it.

## The Approach:

As a relatively naive approach:

- We can iterate through a and b and take the sum of squares and then the square root of the result to produce c
- We add a + b + c:
  - If the sum is the goal then we break
  - If the sum is greater than the goal then we break and continue with an increment
  - If the sum is less than the goal we continue

## The Code:

```
import math

number_to_find = 1000
a = 0
b = 0
c = 0

#we will never reach the end, but it to prevent it from going infinite
for i in range(1,number_to_find):
    for j in range(i,number_to_find): #we can start from i because we technically already
        calculated the reverse of the pair
```

```
c = math.sqrt(i**2 + j**2)
if int(c) == c: #if it is actually an integer, we will continue
    if i + j + c > number_to_find:
        break
    elif i + j + int(c) == number_to_find:
        a = i
        b = j
        break #don't forget this break, it's crucial, lol

if a != 0: #we can break out if we found the triple
    break

print(a,b,c)
print(a+b+c)
print(a*b*c)
print(a**2+b**2, c**2)
```

---

Revision #2

Created 2025-07-25 16:18:43 UTC by Maxwell

Updated 2025-07-25 16:19:52 UTC by Maxwell