Problem 9. A Pythagorean triplet is a set of three natural numbers, a ¡ b ¡ c, for which,

$$a^2 + b^2 = c^2$$

For example, $3^2 + 4^2 = 9 + 16 = 25 = 5^2$.
There exists exactly one Pythagorean triplet for which $a + b + c = 1000$.
Find the product abc.

Knowledge Required: None.

Solution Outline: We need to find three integers a, b, c $(a < b < c)$ such that $a^2 + b^2 = c^2$ and $a + b + c = 1000$. Now we need to find the maximum value that any of the three can take. Let us find for a, the minimum value b, c can take are 1. So a can be at max 998. Now we can implement a brute-force solution which is having 3 nested *for* loops each for a, b, c. We check if conditions have met, if yes then we print the respective a, b, c. This solution takes $\mathrm{O}(n^3)$ time complexity which is slow. We can improve by running only two next *for* loops each for a, b, now c can be calculated by $1000 - (a + b)$. Now we check if current a, b, c follow the Pythagorean triplet rule. If YES, print the respective abc.

Python Solution

```python
def check_pythagorean_triplet(a, b, c):
    return a*a + b*b == c*c

LIMIT = 999
for a in range(1, LIMIT):
    for b in range(1, LIMIT):
        c = 1000 - (a + b)
        if check_pythagorean_triplet(a, b, c):
            # since there is only one solution
            # we print the product and exit
            print(a * b * c)
            exit(0)
```