Primitive Pythagorean Triple

From https://en.wikipedia.org/wiki/Pythagorean_triple

> A **Pythagorean triple** consists of three positive integers a, b, and c, such that $a2 + b2 = c2$. Such a triple is commonly written (a, b, c), and a well-known example is (3, 4, 5). If (a, b, c) is a Pythagorean triple, then so is (ka, kb, kc) for any positive integer k. A **primitive Pythagorean triple** is one in which a, b and c are coprime (that is, they have no common divisor larger than 1). A triangle whose sides form a Pythagorean triple is called a Pythagorean triangle, and is necessarily a right triangle

As shown in program structure example of this question below. A GCD (Greatest common divisor) function is already given, you must write a function **is_coprime(a, b, c)** and **primitive_Pythagorean_triples(max_len)** as described in the comment.

```
def gcd(a,b):
        while b != 0:
                a,b = b, a%b
        return a

def is_coprime(a, b, c):
        # return whether a, b, and c is coprime or not, return as boolean
        # read a definition of coprime here  https://en.wikipedia.org/wiki/Pythagorean_triple
        ???

def primitive_Pythagorean_triples(max_len):
        # return a list contain sublists of 3 values of a, b, and c
        # a ≤ b ≤ c ≤ max_len
        # each sublists are arrange by c from lowest to greatest number
        # if c has the same value, arrange by a
        # for example, if max_len = 65, a list will be
        # [[3, 4, 5], [5, 12, 13], [8, 15, 17], [7, 24, 25],
        # [20, 21, 29], [12, 35, 37], [9, 40, 41], [28, 45, 53],
        # [11, 60, 61], [16, 63, 65], [33, 56, 65]]
        triple = []
        ???

        return triple

exec(input().strip()) # You must have this function here when submit to Grader
```

Input

Command in Python language to test a function

Output

Return output from a function call in input

Example

Input (from keyboard)	Output (on screen)
`print(is_coprime(2,3,6),is_coprime(2,4,8))`	`True False`
`print(primitive_Pythagorean_triples(10))`	`[[3, 4, 5]]`
`print(primitive_Pythagorean_triples(20))`	`[[3, 4, 5], [5, 12, 13], [8, 15, 17]]`