*Assignment 1: Unit test results.*

*A computer screen with white text

Description automatically generated*

*Assignment 1: Greatest Common Denominator (GCD) - Euclid*

def gcd\_euclid(a, b):

    """

    Calculates the Greatest Common Denominator (GCD) using Euclid's algorithm.

    The Euclid's algorithm is based on the principle that the GCD of two numbers

    does not change if the larger number is replaced by its difference with the smaller number.

    Reference: https://www.w3schools.com/dsa/dsa\_ref\_euclidean\_algorithm.php

    The while loop will continue until 'b' is equal to 0.

    In each iteration, 'a' is replaced with 'b' and 'b' is replaced with the remainder of 'a' divided by 'b'.

    The loop will continue until 'b' is equal to 0, at which point the value of 'a' will be the GCD of the two numbers.

    """

    while b != 0:

        a, b = b, a % b  # Replace a with b and b with the remainder of a divided by b

    return a

*Assignment 1: Greatest Common Denominator (GCD) – Test 1*

import unittest

from timeit import timeit

from assignment\_1\_gcd\_euclid import gcd\_euclid

# FILE: test\_gcd\_euclid.py

class TestGCDEuclid(unittest.TestCase):

    def test\_gcd\_euclid(self):

        # Test cases with known inputs and expected outputs

        self.assertEqual(gcd\_euclid(48, 18), 6)

        self.assertEqual(gcd\_euclid(56, 98), 14)

        self.assertEqual(gcd\_euclid(101, 103), 1)

        self.assertEqual(gcd\_euclid(0, 5), 5)

        self.assertEqual(gcd\_euclid(5, 0), 5)

        self.assertEqual(gcd\_euclid(0, 0), 0)

    def test\_performance(self):

        # Measure performance using timeit

        print("Performance test for gcd\_euclid")

        time\_taken = timeit("gcd\_euclid(48, 18)", globals=globals(), number=1000)

        print(f"Time taken for 1000 iterations of gcd\_euclid(48, 18): {time\_taken} seconds")

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

*Assignment 1: Greatest Common Denominator (GCD) – Math library*

import math

def gcd\_math(a, b):

    """

    Calculates the Greatest Common Denominator (GCD) using the math library.

    """

    return math.gcd(a, b)

*Assignment 1: Greatest Common Denominator (GCD) – Test 2*

import unittest

from timeit import timeit

from assignment\_1\_gcd\_math import gcd\_math

# FILE: test\_gcd\_math.py

class TestGCDMath(unittest.TestCase):

    def test\_gcd\_math(self):

        # Test cases with known inputs and expected outputs

        self.assertEqual(gcd\_math(48, 18), 6)

        self.assertEqual(gcd\_math(56, 98), 14)

        self.assertEqual(gcd\_math(101, 103), 1)

        self.assertEqual(gcd\_math(0, 5), 5)

        self.assertEqual(gcd\_math(5, 0), 5)

        self.assertEqual(gcd\_math(0, 0), 0)

    def test\_performance(self):

        # Measure performance using timeit

        print("Performance test for gcd\_math")

        time\_taken = timeit("gcd\_math(48, 18)", globals=globals(), number=1000)

        print(f"Time taken for 1000 iterations of gcd\_math(48, 18): {time\_taken} seconds")

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()