Lab1

```
import time
import random

def gcd_brute_force(nr1, nr2):
    """ Calculate the GCD of two numbers using brute-force iteration,
       by checking from the minimum of the two numbers down to 1.
    """
    if nr1 == 0:
        return nr2
    if nr2 == 0:
        return nr1

    d = min(nr1,nr2)
    while d > 0:
        if nr1 % d == 0 and nr2 % d == 0:
            return d
        d -= 1

def gcd_euclid_algorithm_recursive(nr1, nr2):
    """ Calculate the GCD of two numbers using the recursive Euclidean algorithm.
    """
    if nr1 == 0:
        return nr2
    return gcd_euclid_algorithm_recursive(nr2 % nr1, nr1)


def gcd_using_repeated_subtractions(nr1, nr2):
    """ Calculate the GCD of two numbers using repeated subtraction."""
    if nr1 == 0:
        return nr2
    if nr2 == 0:
        return nr1

    while nr1 != nr2:
        if nr1 > nr2:
            nr1 -= nr2
        else:
            nr2 -= nr1

    return nr1

sum_brute_force = 0
sum_euclid = 0
```

```
sum_repeated_subtractions = 0

for i in range(1, 11):
    nr1 = random.randint(20, 10000000)
    nr2 = random.randint(20, 10000000)

    print(f"Iteration {i} with values: {nr1}, {nr2}")

    start_time = time.perf_counter()
    brute_force_result = gcd_brute_force(nr1, nr2)
    end_time = time.perf_counter()
    brute_force_time = end_time - start_time
    sum_brute_force += brute_force_time
    print("Result Brute Force:", brute_force_result, " time:", brute_force_time)

    start_time = time.perf_counter()
    euclid_result = gcd_euclid_algorithm_recursive(nr1, nr2)
    end_time = time.perf_counter()
    euclid_time = end_time - start_time
    sum_euclid += euclid_time
    print("Result Euclidean:", euclid_result, " time:", euclid_time)

    start_time = time.perf_counter()
    repeated_subtractions_result = gcd_using_repeated_subtractions(nr1, nr2)
    end_time = time.perf_counter()
    repeated_subtractions_time = end_time - start_time
    sum_repeated_subtractions += repeated_subtractions_time
    print("Result Repeated Subtraction:", repeated_subtractions_result, " time:",
repeated_subtractions_time)
    print()

print("Final results: ")
print("Brute:", sum_brute_force/10)
print("Euclid:", sum_euclid/10)
print("Repeated substractions", sum_repeated_subtractions/10)
```