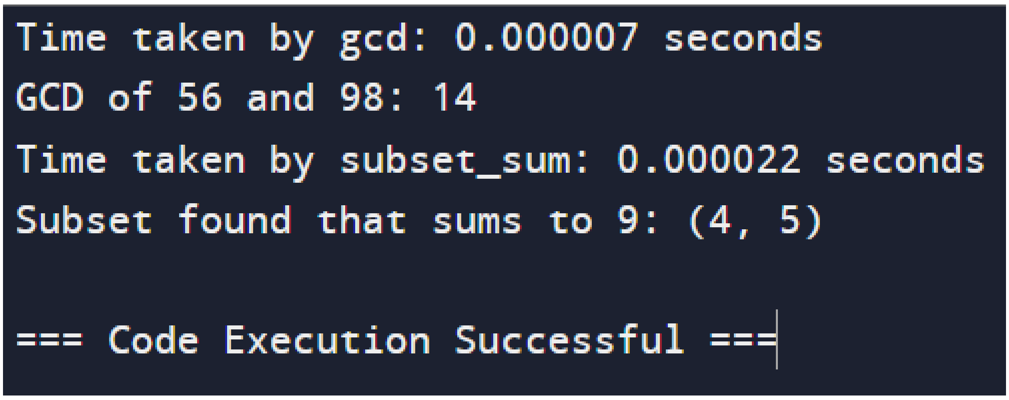
119. Class Problem : P and NP Problems

PROGRAM:-

```
import time
from itertools import chain, combinations

def time_it(func):
    def wrapper(*args, **kwargs):
        start_time = time.time()
        result = func(*args, **kwargs)
        end_time = time.time()
        print(f"Time taken by {func.__name__}: {end_time - start_time:.6f} seconds")
        return result
    return wrapper

# P problem: Finding the greatest common divisor (GCD)
@time_it
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

# NP problem: Subset sum problem
@time_it
def subset_sum(numbers, target_sum):
    all_subsets = chain.from_iterable(combinations(numbers, r) for r in range(len(numbers)+1))
    for subset in all_subsets:
        if sum(subset) == target_sum:
            return subset
    return None

if __name__ == "__main__":
    # Example for P problem
    a = 56
    b = 98
    print(f"GCD of {a} and {b}: {gcd(a, b)}")

    # Example for NP problem
    numbers = [3, 34, 4, 12, 5, 2]
    target_sum = 9
    result = subset_sum(numbers, target_sum)
    if result:
        print(f"Subset found that sums to {target_sum}: {result}")
    else:
        print(f"No subset found that sums to {target_sum}")
```

OUTPUT:-



```
Time taken by gcd: 0.000007 seconds
GCD of 56 and 98: 14
Time taken by subset_sum: 0.000022 seconds
Subset found that sums to 9: (4, 5)

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(2^n)