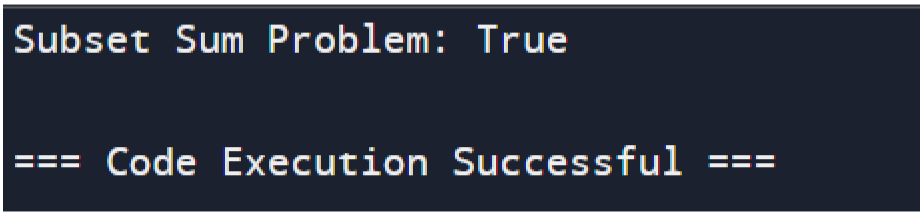
120. NP-Complete and NP-Hard Problem

PROGRAM:-

```
from itertools import combinations

def subset_sum(nums, target):
    """
    Determine if there exists a subset of nums that sums to target.

    Parameters:
    nums (list of int): List of integers.
    target (int): Target sum.

    Returns:
    bool: True if such a subset exists, False otherwise.
    """
    n = len(nums)
    # Check all possible combinations of elements in nums
    for i in range(n + 1):
        for comb in combinations(nums, i):
            if sum(comb) == target:
                return True
    return False

# Example usage:
nums = [3, 34, 4, 12, 5, 2]
target = 9
print("Subset Sum Problem:", subset_sum(nums, target))
```

OUTPUT:-



```
Subset Sum Problem: True


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

TIME COMPLEXITY:-O(2^n)