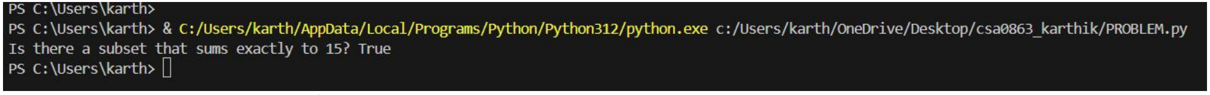
1. 168. Write a program to implement Meet in the Middle Technique. Given a large array of integers and an exact sum E, determine if there is any subset that sums exactly to E. Utilize the Meet in the Middle technique to handle the potentially large size of the array. Return true if there is a subset that sums exactly to E, otherwise return false.
    a) E = {1, 3, 9, 2, 7, 12} exact Sum = 15

Code:

```
from itertools import combinations
def generate_subsets(arr):
   subsets = []
   n = len(arr)
   for i in range(n + 1):
      for combo in combinations(arr, i):
         subsets.append(sum(combo))
   return subsets
def meet_in_the_middle(arr, exact_sum):
   n = len(arr)
   left_half = arr[:n//2]
   right_half = arr[n//2:]
   left_sums = generate_subsets(left_half)
   right_sums = generate_subsets(right_half)
   right_sums_set = set(right_sums)
   for s in left_sums:
      if exact_sum - s in right_sums_set:
         return True
   return False
arr = [1, 3, 9, 2, 7, 12]
exact_sum = 15
output = meet_in_the_middle(arr, exact_sum)
print(f"Is there a subset that sums exactly to {exact_sum}? {output}")
```

output:



```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/csa0863_karthik/PROBLEM.py
Is there a subset that sums exactly to 15? True
PS C:\Users\karth> 
```

Time complexity:f(n)=o($2^{n/2}$)