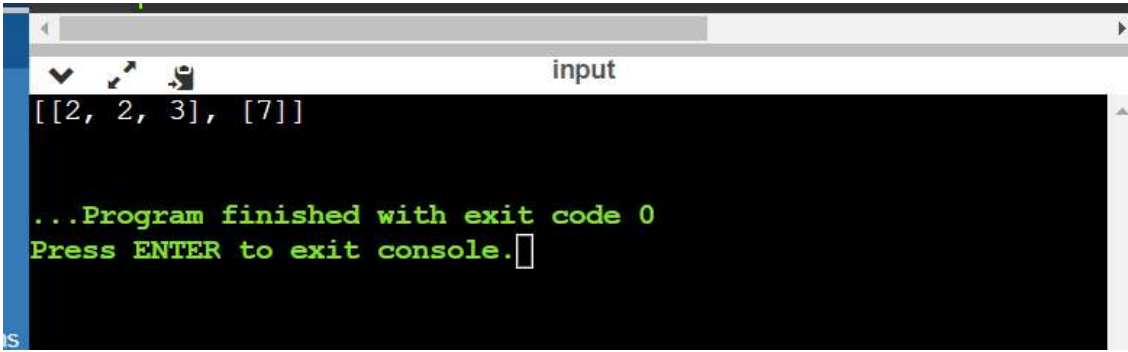


Problem 3: Given an array of distinct integers and a **target**, you have to return *the list of all unique combinations where the chosen numbers sum to target*. You may return the combinations in any order. The same number may be chosen from the given array an unlimited number of times. Two combinations are unique if the frequency of at least one of the chosen numbers is different. It is guaranteed that the number of unique combinations that sum up to **target** is less than 150 combinations for the given input.

```
def combinationSum(candidates, target):  
    results = []  
    backtrack(candidates, target, [], results)  
    return results  
  
def backtrack(candidates, target, combination, results):  
    if target < 0:  
        return  
    if target == 0:  
        results.append(combination)  
        return  
    for i in range(len(candidates)):  
        num = candidates[i]  
        backtrack(candidates[i:], target - num, combination + [num], results)  
  
array = [2, 3, 6, 7]  
target = 7  
result = combinationSum(array, target)  
print(result)
```



The screenshot shows a terminal window with a title bar that says "input". The terminal output displays the result of the function call: `[[2, 2, 3], [7]]`. Below this, a green message states "...Program finished with exit code 0" and another green prompt says "Press ENTER to exit console." with a cursor icon.