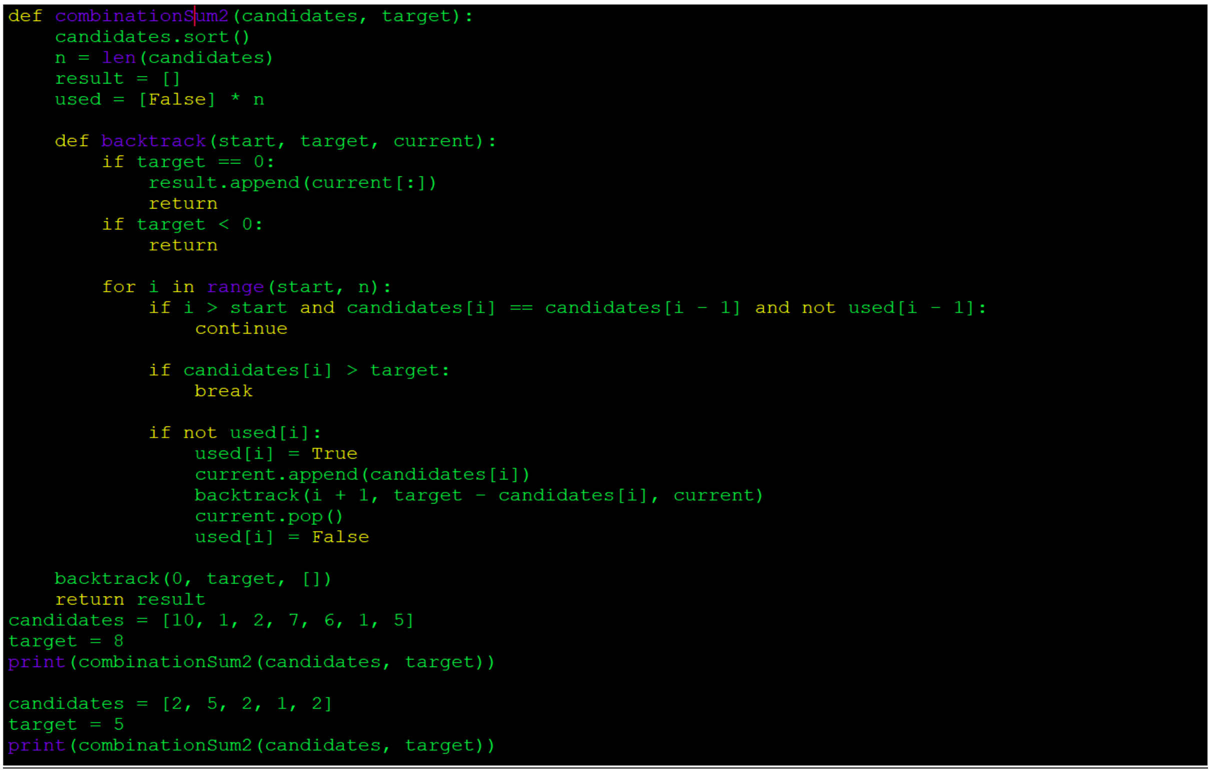
67. . Combination Sum II

Given a collection of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sum to target.

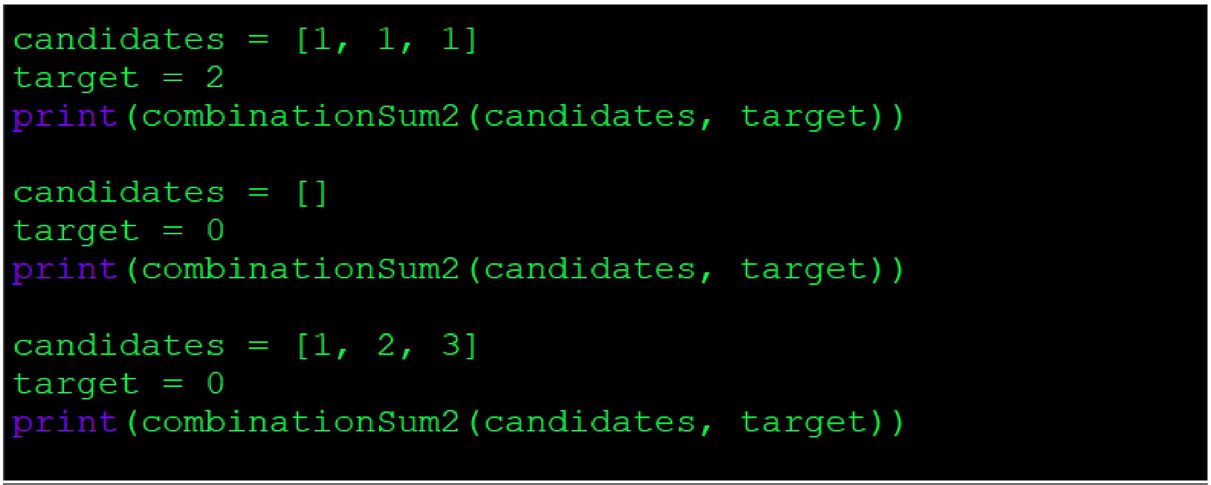
Each number in candidates may only be used once in the combination.

Note: The solution set must not contain duplicate combinations.

Code:

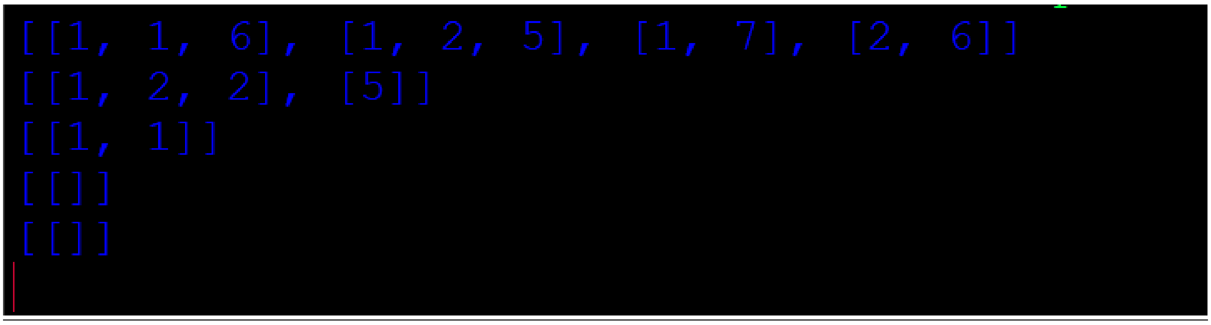


```
def combinationSum2(candidates, target):
    candidates.sort()
    n = len(candidates)
    result = []
    used = [False] * n

    def backtrack(start, target, current):
        if target == 0:
            result.append(current[:])
            return
        if target < 0:
            return

        for i in range(start, n):
            if i > start and candidates[i] == candidates[i - 1] and not used[i - 1]:
                continue

            if candidates[i] > target:
                break

            if not used[i]:
                used[i] = True
                current.append(candidates[i])
                backtrack(i + 1, target - candidates[i], current)
                current.pop()
                used[i] = False

    backtrack(0, target, [])
    return result
candidates = [10, 1, 2, 7, 6, 1, 5]
target = 8
print(combinationSum2(candidates, target))

candidates = [2, 5, 2, 1, 2]
target = 5
print(combinationSum2(candidates, target))
```



```
candidates = [1, 1, 1]
target = 2
print(combinationSum2(candidates, target))

candidates = []
target = 0
print(combinationSum2(candidates, target))

candidates = [1, 2, 3]
target = 0
print(combinationSum2(candidates, target))
```

Output:



```
[[1, 1, 6], [1, 2, 5], [1, 7], [2, 6]]
[[1, 2, 2], [5]]
[[1, 1]]
[[]]
[[]]
```

Time Complexity:

- T(n)= O(nlogn)