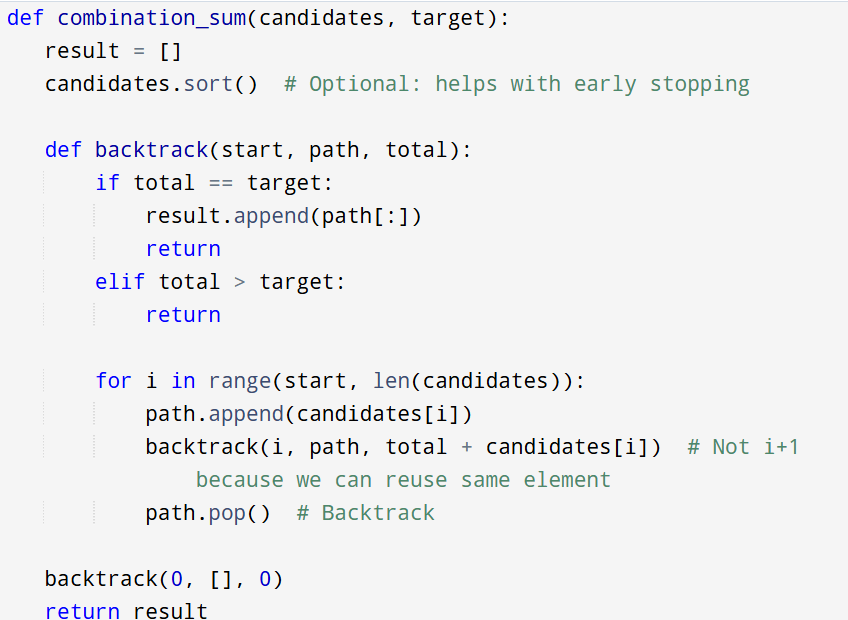
**6.7 Combination Sum**

**Aim:** To construct a python code to solve thenCombination Sum problem.

**Algorithm:**

1. Sort the candidates array to help with pruning.
2. Use a **recursive backtracking function** to explore all combinations.
3. At each recursive call:
4. If the current sum equals target, save the current combination.
5. If the sum exceeds target, backtrack.
6. For each candidate:
7. Choose it, recurse with reduced target.
8. Do not move to the next index (since the same number can be reused).
9. Ensure combinations are built in non-decreasing order to avoid duplicates.

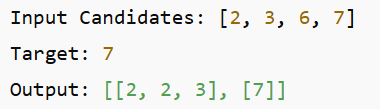
**Program:**



**Input :**  candidates = [2, 3, 6, 7]

target = 7

**Output:**

****

**Result:** Program is been executed.

**Performance analysis:**

* Time complexity: O(2^t)
* Space complexity: O(t)