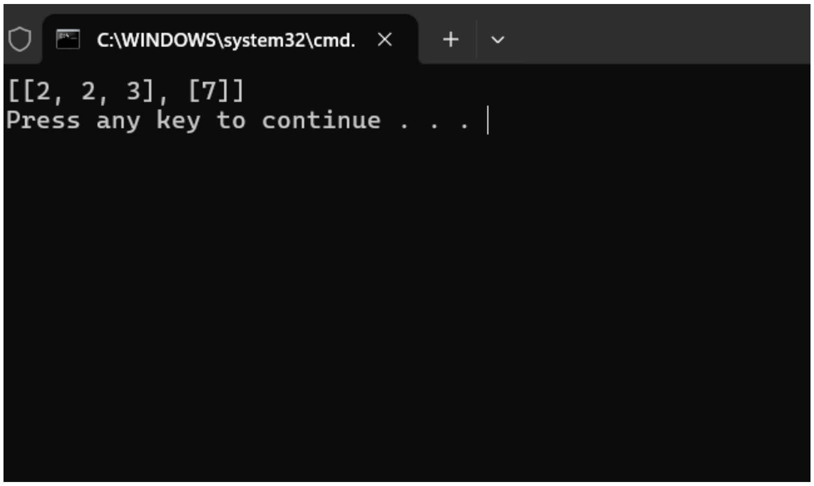
66) Combination Sum

Given an array of distinct integers candidates and a target integer target, return *a list of all unique combinations of* candidates *where the chosen numbers sum to* target. You may return the combinations in any order.
The same number may be chosen from candidates an unlimited number of times. Two combinations are unique if the frequency of at least one of the chosen numbers is different

CODE:

```
def combinationSum(candidates, target):
    candidates.sort()
    result = []

    def backtrack(start, target, current_combination):
        if target == 0:
            result.append(list(current_combination))
            return
        if target < 0:
            return

        for i in range(start, len(candidates)):
            if candidates[i] > target:
                break
            current_combination.append(candidates[i])
            backtrack(i, target - candidates[i], current_combination)
            current_combination.pop()

    backtrack(0, target, [])
    return result

a=[2,3,6,7]
b=7
print(combinationSum(a,b))
```

OUTPUT:



```
[[2, 2, 3], [7]]
Press any key to continue . . .
```

TIME COMPLEXITY : O(nlogn)