1. Dice Throw Problem

```
def dice_throw(n, m, x):
    dp = [[0] * (x + 1) for _ in range(n + 1)]
    dp[0][0] = 1

    for i in range(1, n + 1):
        for j in range(1, x + 1):
            for k in range(1, m + 1):
                if j >= k:
                    dp[i][j] += dp[i-1][j-k]

    return dp[n][x]
n = 3
m = 6
x = 8
print(dice_throw(n, m, x))
```

2. Subset Sum Problem

```
def subset_sum(arr, target):
    n = len(arr)
    dp = [[False] * (target + 1) for _ in range(n + 1)]

    for i in range(n + 1):
        dp[i][0] = True

    for i in range(1, n + 1):
        for j in range(1, target + 1):
            if arr[i-1] <= j:
                dp[i][j] = dp[i-1][j] or dp[i-1][j-arr[i-1]]
            else:
                dp[i][j] = dp[i-1][j]
```

```
    return dp[n][target]
arr = [3, 34, 4, 12, 5, 2]
target = 9
print(subset_sum(arr, target))
```

3. Assembly Line Scheduling

```
def assembly_line(a, t, e, x):
    n = len(a[0])
    dp1 = [0] * n
    dp2 = [0] * n

    dp1[0] = e[0] + a[0][0]
    dp2[0] = e[1] + a[1][0]

    for i in range(1, n):
        dp1[i] = min(dp1[i-1] + a[0][i], dp2[i-1] + t[1][i] + a[0][i])
        dp2[i] = min(dp2[i-1] + a[1][i], dp1[i-1] + t[0][i] + a[1][i])

    return min(dp1[-1] + x[0], dp2[-1] + x[1])
a = [[4, 5, 3, 2], [2, 10, 1, 4]]
t = [[0, 7, 4, 5], [0, 9, 2, 8]]
e = [10, 12]
x = [18, 7]
print(assembly_line(a, t, e, x))
```

4. Longest Palindromic Subsequence

```
def longest_palindromic_subsequence(s):
    n = len(s)
    dp = [[0] * n for _ in range(n)]

    for i in range(n):
        dp[i][i] = 1
```

```
    for cl in range(2, n + 1):
        for i in range(n - cl + 1):
            j = i + cl - 1
            if s[i] == s[j] and cl == 2:
                dp[i][j] = 2
            elif s[i] == s[j]:
                dp[i][j] = dp[i + 1][j - 1] + 2
            else:
                dp[i][j] = max(dp[i][j - 1], dp[i + 1][j])

    return dp[0][n - 1]
s = "bbbab"
print(longest_palindromic_subsequence(s))
```