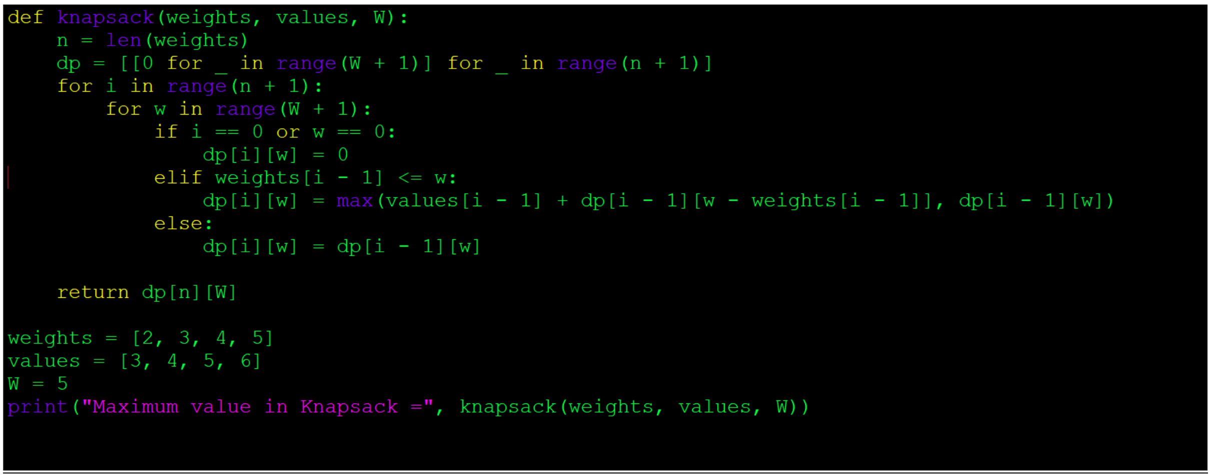
90. Knapsack Problem

Code:

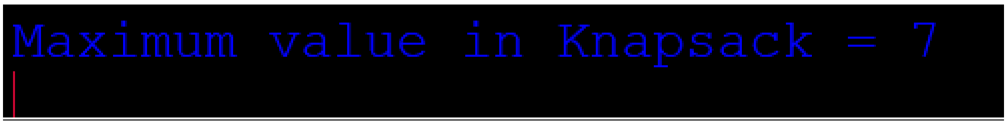


```
def knapsack(weights, values, W):
    n = len(weights)
    dp = [[0 for _ in range(W + 1)] for _ in range(n + 1)]
    for i in range(n + 1):
        for w in range(W + 1):
            if i == 0 or w == 0:
                dp[i][w] = 0
            elif weights[i - 1] <= w:
                dp[i][w] = max(values[i - 1] + dp[i - 1][w - weights[i - 1]], dp[i - 1][w])
            else:
                dp[i][w] = dp[i - 1][w]

    return dp[n][W]

weights = [2, 3, 4, 5]
values = [3, 4, 5, 6]
W = 5
print("Maximum value in Knapsack =", knapsack(weights, values, W))
```

Output:



```
Maximum value in Knapsack = 7
```

Time Complexity:

- T(n)= O(n.w)