89) Knapsack Problem

CODE:

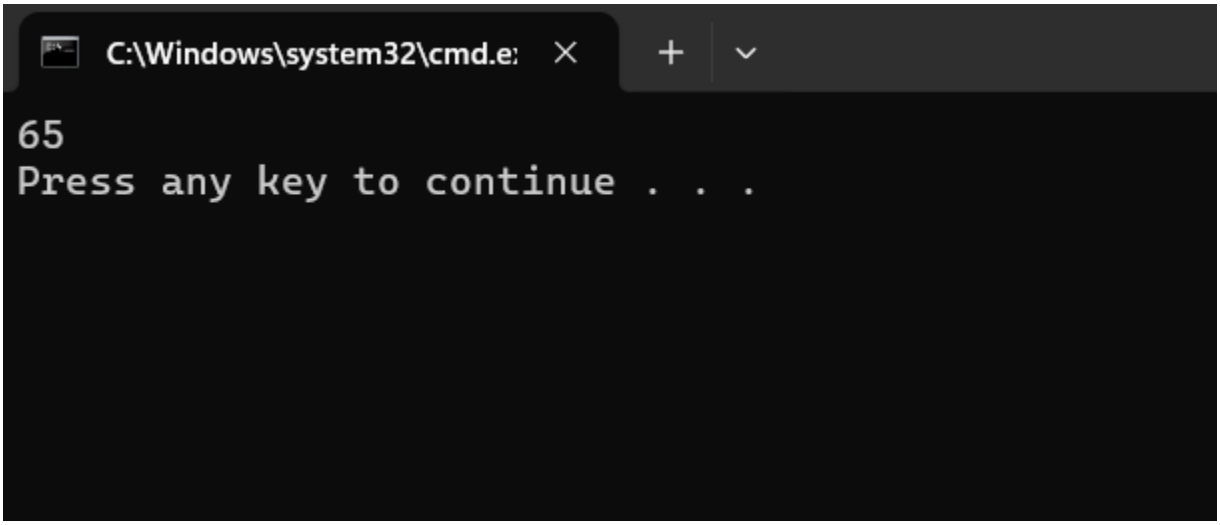
```
def knapsack_01(weights, values, W):
    n = len(weights)
    dp = [[0] * (W + 1) for _ in range(n + 1)]

    for i in range(1, n + 1):
        for w in range(W + 1):
            if weights[i - 1] <= w:
                dp[i][w] = max(dp[i - 1][w], values[i - 1] + dp[i - 1][w - weights[i
- 1]])
            else:
                dp[i][w] = dp[i - 1][w]

    return dp[n][W]


weights = [1, 2, 3]
values = [10, 15, 40]
W = 6
print(knapsack_01(weights, values, W))
```

OUTPUT :



TIME COMPLEXITY : O(nlogn)