

### **EXPERIMENT NO: 04**

```
def knapsack_dp(weights, values, capacity):  
    n = len(values)  
  
    # Create a 2D array to store the maximum value at each n and capacity  
    dp = [[0 for _ in range(capacity + 1)] for _ in range(n + 1)]  
  
    # Build the dp table  
    for i in range(1, n + 1):  
        for w in range(1, capacity + 1):  
            if weights[i - 1] <= w:  
                dp[i][w] = max(dp[i - 1][w], dp[i - 1][w - weights[i - 1]] + values[i - 1])  
            else:  
                dp[i][w] = dp[i - 1][w]  
  
    # The last cell of dp will have the maximum value  
    return dp[n][capacity]  
  
# Example usage  
weights = [1, 3, 4, 5]  
values = [10, 40, 50, 70]  
capacity = 7  
max_value = knapsack_dp(weights, values, capacity)  
print(f'Maximum value in the knapsack: {max_value}')
```

### **OUTPUT:**

Maximum value in the knapsack: 110