107. Knapsack Problem

AIM: To solve the knapsack problem

PROGRAM:

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

    for i in range(1, n + 1):
        for w in range(capacity + 1):
            if 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][capacity]
weights = [10, 20, 30]
values = [60, 100, 120]
capacity = 50

max_value = knapsack(weights, values, capacity)
print(f"Maximum value that can be put in knapsack: {max_value}")
```

OUTPUT:

```
Maximum value that can be put in knapsack: 220
```

TIME COMPLEXITY: O(n * capacity)