DAA – Assignment no: 04

Write a Python program to solve a 0-1 Knapsack problem using dynamic programming or branch and bound strategy.

Input:

Assignment4.py

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

# Example
weights = [10, 20, 30]
values = [60, 100, 120]
capacity = 50

result = knapsack(weights, values, capacity)
print("Maximum value in Knapsack:", result)

```

Output:

```
PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA\Code file> & D:/Python/python.exe "d:/Tanmay Mohadikar/Sem 7 Practicals/DAA/Code file/Assignment4.py"
Maximum value in Knapsack: 220
PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA\Code file>
```