def knapsack(values, weights, capacity):

n = len(values) # Number of items

# Create a 2D array to store maximum value at each n and capacity

dp = [[0 for \_ in range(capacity + 1)] for \_ in range(n + 1)]

# Build the table dp[][] in bottom-up fashion

for i in range(1, n + 1):

for w in range(1, capacity + 1):

# Check if the weight of the current item i can be included

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 the dp table contains the maximum value

return dp[n][capacity]

# Driver code

if \_\_name\_\_ == "\_\_main\_\_":

# Input for number of items

n = int(input("Enter the number of items: "))

# Input for values and weights

values = []

weights = []

for i in range(n):

value = int(input(f"Enter the value for item {i + 1}: "))

weight = int(input(f"Enter the weight for item {i + 1}: "))

values.append(value)

weights.append(weight)

# Input for the maximum weight capacity of the knapsack

capacity = int(input("Enter the maximum weight capacity of the knapsack: "))

# Function call

max\_val = knapsack(values, weights, capacity)

print(f"The maximum value in the knapsack is: {max\_val}")

-------------------------------------------------------------------------------------------------------------------------------------

Output:

>>> %Run -c $EDITOR\_CONTENT

Enter the number of items: 5

Enter the value for item 1: 2

Enter the weight for item 1: 1

Enter the value for item 2: 6

Enter the weight for item 2: 4

Enter the value for item 3: 5

Enter the weight for item 3: 4

Enter the value for item 4: 8

Enter the weight for item 4: 3

Enter the value for item 5: 4

Enter the weight for item 5: 5

Enter the maximum weight capacity of the knapsack: 10

The maximum value in the knapsack is: 16