def knapsack\_dynamic\_programming(weights, values, capacity):

    n = len(values)

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

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

    # Build the dp array in bottom-up fashion

    for i in range(n + 1):

        for w in range(capacity + 1):

            if i == 0 or w == 0:

                dp[i][w] = 0

            elif 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]

# Taking input from the user

weights = list(map(int, input("Enter the weights of items separated by space: ").split()))

values = list(map(int, input("Enter the values of items separated by space: ").split()))

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

# Calculating and displaying the maximum value

print("Maximum value in Knapsack =", knapsack\_dynamic\_programming(weights, values, capacity))

output-

Enter the weights of items separated by space: 1 2 3 2

Enter the values of items separated by space: 8 4 0 5

Enter the capacity of the knapsack: 5

Maximum value in Knapsack = 13