# Structure for an item which stores weight and corresponding value of Item

class Item:

def \_\_init\_\_(self, profit, weight):

self.profit = profit

self.weight = weight

# Main greedy function to solve the fractional knapsack problem

def fractionalKnapsack(W, arr):

# Sorting items based on the value-to-weight ratio in descending order

arr.sort(key=lambda x: (x.profit / x.weight), reverse=True)

# Result (value in the knapsack)

finalvalue = 0.0

# Looping through all items

for item in arr:

# If adding the item won't overflow, add it completely

if item.weight <= W:

W -= item.weight

finalvalue += item.profit

else:

# If we can't add the current item, add a fractional part of it

finalvalue += item.profit \* (W / item.weight)

break # Knapsack is full

# Returning the final value

return finalvalue

# Driver code

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

# Input for the maximum weight capacity of the knapsack

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

# Input for number of items

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

arr = []

# Input for each item's profit and weight

for i in range(n):

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

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

arr.append(Item(profit, weight)) # Create an Item and add to the list

# Function call

max\_val = fractionalKnapsack(W, arr)

print(f"The maximum value in the knapsack is: {max\_val:.2f}")

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

Output:

>>> %Run -c $EDITOR\_CONTENT

Enter the maximum weight capacity of the knapsack: 5

Enter the number of items: 6

Enter the profit for item 1: 5

Enter the weight for item 1: 8

Enter the profit for item 2: 3

Enter the weight for item 2: 6

Enter the profit for item 3: 4

Enter the weight for item 3: 1

Enter the profit for item 4: 2

Enter the weight for item 4: 5

Enter the profit for item 5: 4

Enter the weight for item 5: 5

Enter the profit for item 6: 1

Enter the weight for item 6: 2

The maximum value in the knapsack is: 7.20