class Item:

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

self.profit = profit

self.weight = weight

def compare(a, b):

return a.profit / a.weight > b.profit / b.weight

def fractionalKnapsack(W, items):

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

maxValue = 0.0

for item in items:

if W == 0:

break

if item.weight <= W:

maxValue += item.profit

W -= item.weight

else:

maxValue += item.profit \* W / item.weight

break

return maxValue

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

W = int(input("Enter knapsack capacity: "))

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

items = []

for i in range(n):

profit, weight = map(int, input(f"Enter profit and weight for item {i + 1}: ").split())

items.append(Item(profit, weight))

result = fractionalKnapsack(W, items)

print(f"Maximum value: {result}")

output:-

Enter knapsack capacity: 50

Enter the number of items: 3

Enter profit and weight for item 1: 60 10

Enter profit and weight for item 2: 100 20

Enter profit and weight for item 3: 120 30

Maximum value: 240.0