class Item:

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

        self.value = value

        self.weight = weight

def fractional\_knapsack(items, capacity):

    # Calculate value to weight ratio for each item and sort them in descending order

    items.sort(key=lambda item: item.value/item.weight, reverse=True)

    total\_value = 0.0

    for item in items:

        if capacity > 0 and item.weight <= capacity:

            # If the item can be fully taken, take it

            capacity -= item.weight

            total\_value += item.value

        else:

            # If the item cannot be fully taken, take the fractional part

            fraction = capacity / item.weight

            total\_value += item.value \* fraction

            break

    return total\_value

def main():

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

    items = []

    for i in range(n):

        value = float(input(f"Enter value of item {i + 1}: "))

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

        items.append(Item(value, weight))

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

    max\_value = fractional\_knapsack(items, capacity)

    print(f"The maximum value of items that can be carried: {max\_value}")

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

    main()

Output:

Enter the number of items: 3

Enter value of item 1: 60

Enter weight of item 1: 10

Enter value of item 2: 100

Enter weight of item 2: 20

Enter value of item 3: 120

Enter weight of item 3: 30

Enter the capacity of the knapsack: 50

The maximum value of items that can be carried: 240.0