

```

class Item:
    def __init__(self, value, weight):
        self.value = value
        self.weight = weight
        self.ratio = value / weight # profit-to-weight ratio

def fractional_knapsack(capacity, items):
    # Sort items by their ratio in descending order
    items.sort(key=lambda x: x.ratio, reverse=True)

    total_value = 0.0 # Total value accumulated in the knapsack

    for item in items:
        if capacity == 0:
            break # No more capacity left in the knapsack

        if item.weight <= capacity:
            # If the item can fit in the knapsack, take it whole
            capacity -= item.weight
            total_value += item.value
        else:
            # If the item can't fit, take the fraction that fits
            total_value += item.ratio * capacity
            capacity = 0 # The knapsack is now full

    return total_value

# Main program
if __name__ == "__main__":
    n = int(input("Enter number of items: "))
    items = []

    for i in range(n):
        value = int(input(f"Enter value of item {i + 1}: "))
        weight = int(input(f"Enter weight of item {i + 1}: "))
        items.append(Item(value, weight))

    capacity = int(input("Enter maximum weight capacity of the knapsack: "))
    max_value = fractional_knapsack(capacity, items)

    print(f'The maximum value that can be carried in the knapsack: {max_value:.2f}')

```