

Practical- 2 DAA

Name- Pratham Mahabare

Roll No.- 39

Code-

```
def fractional_knapsack(values, weights, capacity):
    """
    Returns (total_value, selection)
    where selection is a list of tuples (index, taken_weight, taken_value, fraction_taken)
    """
    if len(values) != len(weights):
        raise ValueError("values and weights must have same length")
    if capacity <= 0:
        return 0.0, []

    items = []
    for i, (v, w) in enumerate(zip(values, weights)):
        if w <= 0:
            raise ValueError(f"Weight must be positive for item {i}, got {w}")
        ratio = v / w
        items.append((ratio, i, v, w))

    # Sort by ratio descending
    items.sort(key=lambda x: x[0], reverse=True)

    remaining = capacity
    total_value = 0.0
    selection = []

    for ratio, idx, val, wt in items:
        if remaining <= 0:
            break
        if wt <= remaining:
            # take whole item
            total_value += val
            selection.append((idx, wt, val, 1.0))
            remaining -= wt
        else:
            # take fraction
            fraction = remaining / wt
            taken_value = val * fraction
            total_value += taken_value
            selection.append((idx, remaining, taken_value, fraction))
            remaining = 0
            break

    return total_value, selection
```

```

# Example interactive usage
if __name__ == "__main__":
    try:
        n = int(input("Enter number of items: "))
        values = [float(i) for i in input("Enter values (space separated): ").split()]
        weights = [float(i) for i in input("Enter weights (space separated): ").split()]
        capacity = float(input("Enter maximum capacity: "))

        total, sel = fractional_knapsack(values, weights, capacity)
        print(f"\nMaximum value (fractional knapsack): {total:.4f} units")
        print("Selection (index, taken_weight, taken_value, fraction):")
        for s in sel:
            print(s)
    except KeyboardInterrupt:
        print("\nTerminated.")
    except Exception as e:
        print("Error:", e)

```

Output-

```

pratham_mahabare@fedora:~/Downloads/LP-3/DAA$ python3 2_knapsack.py
Enter number of items: 3
Enter values (space separated): 60 100 120
Enter weights (space separated): 10 20 30
Enter maximum capacity: 50

Maximum value (fractional knapsack): 240.0000 units
Selection (index, taken_weight, taken_value, fraction):
(0, 10.0, 60.0, 1.0)
(1, 20.0, 100.0, 1.0)
(2, 20.0, 80.0, 0.6666666666666666)
pratham_mahabare@fedora:~/Downloads/LP-3/DAA$ 

```