Name : Darshan Bele

Div : A , Batch : A1

Practical 3

```
from tabulate import tabulate
def fractional_knapsack(items, capacity):
    # Sort items by value/weight ratio in descending order
    items.sort(key=lambda x: x['profit'] / x['weight'], reverse=True)
    total_value = 0.0
    knapsack_items = []
    remaining_capacity = capacity

    for item in items:
        if remaining_capacity == 0:
            break
        if item['weight'] <= remaining_capacity:
            # Take full item
            total_value += item['profit']
            knapsack_items.append({
                'id': item['id'],
                'total_profit': item['profit'],
                'total_weight': item['weight'],
                'fraction': 1.0,
                'weight_taken': item['weight'],
                'profit_gained': item['profit']
            })
            remaining_capacity -= item['weight']
        else:
            # Take fractional part
            fraction = remaining_capacity / item['weight']
            profit_gained = item['profit'] * fraction
            weight_taken = remaining_capacity
            total_value += profit_gained
            knapsack_items.append({
                'id': item['id'],
                'total_profit': item['profit'],
                'total_weight': item['weight'],
                'fraction': fraction,
                'weight_taken': weight_taken,
                'profit_gained': profit_gained
            })
            remaining_capacity = 0

    return total_value, knapsack_items

# --------- USER INPUT SECTION ---------
n = 6
items = []
for i in range(1, n+1):
    profit = float(input(f"Enter profit for item {i}: "))
    while True:
        weight = float(input(f"Enter weight for item {i} (must be > 0): "))
        if weight > 0:
            break
        print("Weight must be greater than zero. Please enter again.")
    items.append({'id': i, 'profit': profit, 'weight': weight})

capacity = float(input("\nEnter knapsack capacity: "))

# --------- DISPLAY INPUT TABLE ---------
print("\nInput Items Table:")
input_table = [
    [item['id'], item['profit'], item['weight'], round(item['profit'] / item['weight'], 2)]
    for item in items
]
print(tabulate(input_table, headers=["Item ID", "Profit", "Weight", "Profit/Weight"], tablefmt="fancy_grid"))

# --------- SOLVE THE PROBLEM ---------
max_value, taken_items = fractional_knapsack(items, capacity)

# --------- DISPLAY OUTPUT TABLE WITH CALCULATIONS ---------
print("\nItems Taken in the Knapsack (with calculations):")
```

```
output_table = [
    [
        item['id'],
        item['total_profit'],
        item['total_weight'],
        f"{item['fraction']:.2f}",
        f"{item['weight_taken']:.2f}",
        f"{item['profit_gained']:.2f}"
    ]
    for item in taken_items
]
print(tabulate(output_table, headers=[
    "Item ID", "Total Profit", "Total Weight", "Fraction Taken",
    "Weight Taken", "Profit Gained"
], tablefmt="fancy_grid"))

print(f"\nMaximum Profit: {max_value:.2f}")
```

OUTPUT :

```
PS C:\Users\darsh\Desktop\DAA]>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\darsh\.vscode\extensions\ms-
python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '54767' '--' 'c:\Users\darsh\Desktop\DAA]\pract3'
Enter profit for item 1: 6
Enter weight for item 1 (must be > 0): 10
Enter profit for item 2: 100
Enter weight for item 2 (must be > 0): 20
Enter profit for item 3: 120
Enter weight for item 3 (must be > 0): 30
Enter profit for item 4: 80
Enter weight for item 4 (must be > 0): 40
Enter profit for item 5: 30
Enter weight for item 5 (must be > 0): 10
Enter profit for item 6: 50
Enter weight for item 6 (must be > 0): 20

Enter knapsack capacity: 50
```

Input Items Table:

```
╒══════════╤══════════╤══════════╤════════════════╕
│ Item ID │ Profit │ Weight │ Profit/Weight │
╞══════════╪══════════╪══════════╪════════════════╡
│      1 │     6 │    10 │        0.6 │
├──────────┼──────────┼──────────┼────────────────┤
│      2 │   100 │    20 │          5  │
├──────────┼──────────┼──────────┼────────────────┤
│      3 │   120 │    30 │          4  │
├──────────┼──────────┼──────────┼────────────────┤
│      4 │    80 │    40 │          2  │
├──────────┼──────────┼──────────┼────────────────┤
│      5 │    30 │    10 │          3  │
├──────────┼──────────┼──────────┼────────────────┤
════╤══════════════════╕
│ Item ID │ Total Profit │ Total Weight │ Fraction Taken │ Weight Taken │ Profit Gained │
╞══════════╪════════════════╪════════════════╪══════════════════╪═══════
═════════╪═════════════════╡
│      2 │        100 │          20 │            1 │         20 │          100 │
├──────────┼────────────────┼────────────────┼──────────────────┼───────
─────────┼─────────────────┤
│      3 │        120 │          30 │            1 │         30 │          120 │
╘══════════╧════════════════╧════════════════╧══════════════════╧═══════
═════════╧═════════════════╛
```

Maximum Profit: 220.00