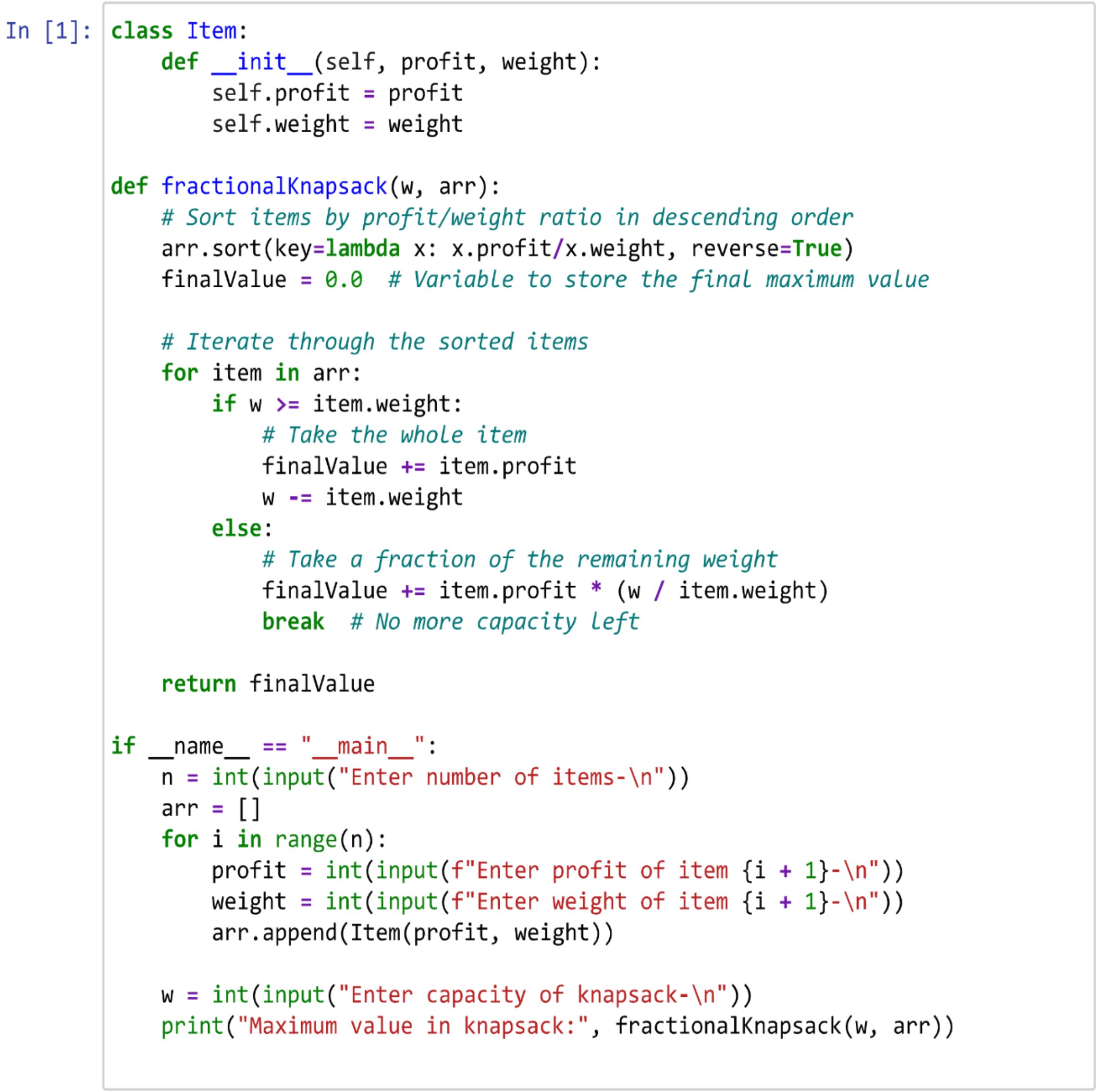
Name: Vaishnavi Pravin Kolse

Class: BE - A

Roll No.37

Practical No.3

Aim: Write a program to solve a fractional Knapsack problem using a greedy method.



```
In [1]: class Item:
            def __init__(self, profit, weight):
                self.profit = profit
                self.weight = weight

        def fractionalKnapsack(w, arr):
            # Sort items by profit/weight ratio in descending order
            arr.sort(key=lambda x: x.profit/x.weight, reverse=True)
            finalValue = 0.0  # Variable to store the final maximum value

            # Iterate through the sorted items
            for item in arr:
                if w >= item.weight:
                    # Take the whole item
                    finalValue += item.profit
                    w -= item.weight
                else:
                    # Take a fraction of the remaining weight
                    finalValue += item.profit * (w / item.weight)
                    break  # No more capacity left

            return finalValue

        if __name__ == "__main__":
            n = int(input("Enter number of items-\n"))
            arr = []
            for i in range(n):
                profit = int(input(f"Enter profit of item {i + 1}-\n"))
                weight = int(input(f"Enter weight of item {i + 1}-\n"))
                arr.append(Item(profit, weight))

            w = int(input("Enter capacity of knapsack-\n"))
            print("Maximum value in knapsack:", fractionalKnapsack(w, arr))
```

```
Enter number of items-
5
Enter profit of item 1-
30
Enter weight of item 1-
5
Enter profit of item 2-
40
Enter weight of item 2-
10
Enter profit of item 3-
45
Enter weight of item 3-
15
Enter profit of item 4-
77
Enter weight of item 4-
22
Enter profit of item 5-
90
Enter weight of item 5-
25
Enter capacity of knapsack-
60
Maximum value in knapsack: 230.0
```

In [ ]: