Program:

```
# Class to represent an item in the Knapsack
class Item:
    def __init__(self, value, weight):
        self.value = value
        self.weight = weight
        self.ratio = value / weight

# Function to solve the Fractional Knapsack problem using Greedy strategy
def fractional_knapsack(capacity, items):
    # Sort items by value-to-weight ratio in decreasing order
    items.sort(key=lambda x: x.ratio, reverse=True)

    total_value = 0  # Total value of knapsack
    for item in items:
        if capacity >= item.weight:
            # If the item can be fully taken, take it
            capacity -= item.weight
            total_value += item.value
        else:
            # Take the fraction of the remaining item
            total_value += item.value * (capacity / item.weight)
            break

    return total_value

# Input
n = int(input("Enter the 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 the capacity of the knapsack: "))

# Perform the greedy strategy to solve the knapsack problem
max_value = fractional_knapsack(capacity, items)
print(f"Maximum value in Knapsack = {max_value}")
```

Output: