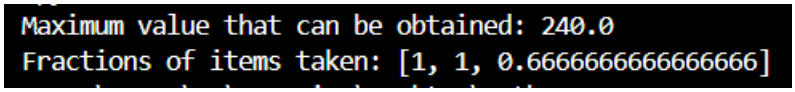
89. Knapsack Problem

AIM: To solve the Knapsack Problem by using greedy method

PROGRAM:

```
def knapsack_greedy(values, weights, capacity):
    n = len(values)
    index = list(range(n))
    ratio = [v/w for v, w in zip(values, weights)]
    index.sort(key=lambda i: ratio[i], reverse=True)

    max_value = 0
    fractions = [0]*n

    for i in index:
        if weights[i] <= capacity:
            fractions[i] = 1
            max_value += values[i]
            capacity -= weights[i]
        else:
            fractions[i] = capacity / weights[i]
            max_value += values[i] * fractions[i]
            break

    return max_value, fractions

values = [60, 100, 120]
weights = [10, 20, 30]
capacity = 50

max_value, fractions = knapsack_greedy(values, weights, capacity)
print("Maximum value that can be obtained:", max_value)
```

print("Fractions of items taken:", fractions)

OUTPUT: 

TIME COMPLEXITY: O (nlog n)