**Code:-**

def fractional\_knapsack(value, weight, capacity):

# index = [0, 1, 2, ..., n - 1] for n items

index = list(range(len(value)))

# contains ratios of values to weight

ratio = [v / w for v, w in zip(value, weight)]

# index is sorted according to value-to-weight ratio in decreasing order

index.sort(key=lambda i: ratio[i], reverse=True)

max\_value = 0

fractions = [0] \* len(value)

for i in index:

if weight[i] <= capacity:

fractions[i] = 1

max\_value += value[i]

capacity -= weight[i]

else:

fractions[i] = capacity / weight[i]

max\_value += value[i] \* capacity / weight[i]

break

return max\_value, fractions

if \_\_name\_\_ == '\_\_main\_\_':

n = int(input('Enter number of items: '))

value = input('Enter the values of the {} item(s) in order: '.format(n)).split()

value = [int(v) for v in value]

weight = input('Enter the positive weights of the {} item(s) in order: '.format(n)).split()

weight = [int(w) for w in weight]

capacity = int(input('Enter maximum weight: '))

max\_value, fractions = fractional\_knapsack(value, weight, capacity)

print('The maximum value of items that can be carried:', max\_value)

print('The fractions in which the items should be taken:', fractions)

**Output:-**

C:\Users\asus\PycharmProjectsCommunity\LP3\venv\Scripts\python.exe "F:\7th Sem\LP3 Practical\DAA\_FInal\3\_Fractinal\_Knapsack\Fractional Knapsack Problem.py"

Enter number of items: 3

Enter the values of the 3 item(s) in order: 15 24 25

Enter the positive weights of the 3 item(s) in order: 10 15 18

Enter maximum weight: 20

The maximum value of items that can be carried: 31.5

The fractions in which the items should be taken: [0.5, 1, 0]

Process finished with exit code 0