**Code: aass**

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

"""Return maximum value of items and their fractional amounts.

(max\_value, fractions) is returned where max\_value is the maximum value of

items with total weight not more than capacity.

fractions is a list where fractions[i] is the fraction that should be taken

of item i, where 0 <= i < total number of items.

value[i] is the value of item i and weight[i] is the weight of item i

for 0 <= i < n where n is the number of items.

capacity is the maximum weight.

"""

# 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

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)