```
#Knapsack
# Smallest Weight
def Knapsack(wt, profit, cap):
    items = [(i, wt[i], profit[i], wt[i]/profit[i]) for i in range(len(wt))]
    items.sort(key= lambda x:x[1])

    total_profit = 0
    include_items = []

    for i, wt, profit, ratio in items:
        if cap >= wt:
            include_items.append((i, 1))
            total_profit = total_profit + profit
            cap = cap - wt
        else:
            frac = cap/wt
            total_profit = total_profit + (frac * profit)
            include_items.append((i, frac))

    return total_profit, include_items

wt = [30, 20, 10]
profit = [60, 50, 40]
cap = 50

total_profit, include_items = Knapsack(wt, profit, cap)
print("\nTotal profit is: ", total_profit)
print("Include items are: ", include_items)

# Largest Ratio
def Knapsack(wt, profit, cap):
    items = [(i, wt[i], profit[i], wt[i]/profit[i]) for i in range(len(wt))]
    items.sort(key= lambda x:x[3], reverse=True)

    total_profit = 0
    include_items = []
```

```
    for i, wt, profit, ratio in items:
        if cap >= wt:
            include_items.append((i, 1))
            total_profit = total_profit + profit
            cap = cap - wt
        else:
            frac = cap/wt
            total_profit = total_profit + (frac * profit)
            include_items.append((i, frac))

    return total_profit, include_items

wt = [30, 20, 10]
profit = [60, 50, 40]
cap = 50

total_profit, include_items = Knapsack(wt, profit, cap)
print("\nTotal profit is: ", total_profit)
print("Include items are: ", include_items)

# Largest Profit
def Knapsack(wt, profit, cap):
    items = [(i, wt[i], profit[i], wt[i]/profit[i]) for i in range(len(wt))]
    items.sort(key= lambda x:x[2], reverse=True)

    total_profit = 0
    include_items = []

    for i, wt, profit, ratio in items:
        if cap >= wt:
            include_items.append((i, 1))
            total_profit = total_profit + profit

            cap = cap - wt
        else:
            frac = cap/wt
            total_profit = total_profit + (frac * profit)
            include_items.append((i, frac))
```

```
    return total_profit, include_items

wt = [30, 20, 10]
profit = [60, 50, 40]
cap = 50

total_profit, include_items = Knapsack(wt, profit, cap)
print("\nTotal profit is: ", total_profit)
print("Include items are: ", include_items)
```

Output :

```
Total profit is:  130.0
Include items are:  [(2, 1), (1, 1), (0, 0.6666666666666666)]

Total profit is:  110.0
Include items are:  [(0, 1), (1, 1), (2, 0.0)]

Total profit is:  110.0
Include items are:  [(0, 1), (1, 1), (2, 0.0)]
```