package lp3;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

class Item {

int weight;

int value;

double valuePerWeight;

public Item(int weight, int value) {

this.weight = weight;

this.value = value;

this.valuePerWeight = (double) value / weight;

}

}

public class Knapsack\_problem {

public static double fractionalKnapsack(int[] weights, int[] values, int capacity) {

int n = weights.length;

List<Item> itemList = new ArrayList<>();

// Create a list of items with their value-to-weight ratios

for (int i = 0; i < n; i++) {

itemList.add(new Item(weights[i], values[i]));

}

// Sort items by value-to-weight ratio in descending order

Collections.sort(itemList, Comparator.comparingDouble((Item item) -> item.valuePerWeight).reversed());

double maxValue = 0;

int currentWeight = 0;

// Fill the knapsack with items fractionally

for (Item item : itemList) {

if (currentWeight + item.weight <= capacity) {

maxValue += item.value;

currentWeight += item.weight;

} else {

int remainingCapacity = capacity - currentWeight;

maxValue += item.valuePerWeight \* remainingCapacity;

break; // Knapsack is full

}

}

return maxValue;

}

public static void main(String[] args) {

int[] weights = {18, 15, 20};

int[] values = {24, 25, 15};

int capacity = 20;

double maxValue = fractionalKnapsack(weights, values, capacity);

System.out.println("Maximum value obtained = " + maxValue);

}

}