// DAA Practical 3 - Fractional Knapsack Problem

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

class Item {

int value;

int weight;

double ratio;

public Item(int value, int weight, double ratio) {

this.value = value;

this.weight = weight;

this.ratio = ratio;

}

}

public class Main {

public static double fracKnapsack(List<Item> items, int limit) {

Collections.sort(items, Comparator.comparing(Item::ratio).reversed());

double totalValue = 0.0;

int currentWeight = 0;

for (Item item : items) {

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

totalValue += item.value;

currentWeight += item.weight;

} else {

double remainingWeight = limit - currentWeight;

totalValue += item.ratio \* remainingWeight;

break;

}

}

return totalValue;

}

public static void main(String[] args) {

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

items.add(new Item(60, 10, 0.0));

items.add(new Item(100, 20, 0.0));

items.add(new Item(120, 30, 0.0));

int limit = 50;

for (Item item : items) {

item.ratio = (double) item.value / item.weight;

}

double profit = fracKnapsack(items, limit);

System.out.println(profit);

}

}