**package** GreedyAlgorithm;

**import** java.util.ArrayList;

**import** java.util.Collections;

**class** ItemProfitWeightRatio **implements** Comparable<ItemProfitWeightRatio>

{

**public int itemNo**;

**public int profit**;

**public int weight**;

**public double ratio**;

**public** ItemProfitWeightRatio(**int** itemNo, **int** profit, **int** weight, **double** ratio) {

**this**.**itemNo** = itemNo;

**this**.**profit** = profit;

**this**.**weight** = weight;

**this**.**ratio** = ratio;

}

@Override

**public int** compareTo(ItemProfitWeightRatio o) {

**if** (**this**.**ratio**<o.**ratio**) **return** 1;

**else return** -1;

}

}

**class** FindMaxProfit

{

ArrayList<ItemProfitWeightRatio> **arrayList**;

**public int maxCapacity**;

**public int**[] **object**;

**public double**[] **profitCalculate**;

**public int index**;

**public double totalProfit**;

**public** FindMaxProfit(ArrayList<ItemProfitWeightRatio> arrayList, **int** maxCapacity, **int**[] object) {

**this**.**arrayList** = arrayList;

**this**.**maxCapacity** = maxCapacity;

**this**.**object** = object;

**this**.**profitCalculate** = **new double**[100];

**this**.**totalProfit** = 0;

giveItems();

}

**public void** giveItems()

{

**for** (ItemProfitWeightRatio give : **arrayList**)

{

checkWeight(give.**itemNo**, give.**profit**, give.**weight**, give.**ratio**);

}

}

**public void** checkWeight(**int** itemName, **int** profit, **int** weight, **double** ratio)

{

**maxCapacity** = **maxCapacity** - weight;

**if** (**maxCapacity** == 0)

{

**profitCalculate**[**index**] = profit;

**object**[**index**] = itemName;

**index** = **index** + 1;

**return**;

}

**else if** (**maxCapacity**<0)

{

**maxCapacity** = **maxCapacity** + weight;

**double** result = Double.*valueOf*(profit) \* ((**maxCapacity**)/Double.*valueOf*(weight));

**profitCalculate**[**index**] = result;

**index**++;

**maxCapacity** =0;

**return**;

}

**else**

{

**profitCalculate**[**index**] = profit;

**object**[**index**] = itemName;

**index** = **index** + 1;

}

}

**public void** totalProfit()

{

**for** (**int** i=0;i<**index**;i++)

{

**if** (**profitCalculate**[i]==0){}**else** {

System.***out***.println(**"item No: "** + **object**[i] + **" "** + **"profit = "** + **profitCalculate**[i]);

**totalProfit** = **totalProfit** + **profitCalculate**[i];

}

}

System.***out***.println(**"-----------------------------"**);

System.***out***.println(**"Total Profit = "**+**totalProfit**);

}

}

**public class** FractionalKnapsackUsingRatio {

**public static void** main(String[] args) {

**int**[] itemName = {1,2,3,4,5,6,7};

**int**[] profit = {5,10,15,7,8,9,4};

**int**[] weight = {1,3,5,4,1,3,2};

**double**[] ratio = **new double**[100];

**int** maxCapacity = 15;

ArrayList<ItemProfitWeightRatio> arrayList = **new** ArrayList<>();

**for** (**int** i = 0; i <itemName.**length** ; i++)

{

**double** ratioFind = (Double.*valueOf*(profit[i])/Double.*valueOf*(weight[i]));

arrayList.add(**new** ItemProfitWeightRatio(itemName[i],profit[i],weight[i],ratioFind));

}

Collections.*sort*(arrayList);

**int**[] itemNo = **new int**[100];

FindMaxProfit object = **new** FindMaxProfit(arrayList,maxCapacity,itemName);

object.totalProfit();

}

}