**package** GreedyAlgorithmLab;

**import** java.util.ArrayList;

**import** java.util.Collections;

**class** knapsackMaxProfit **implements** Comparable<knapsackMaxProfit>

{

**public int itemName**;

**public int profit**;

**public int weight**;

**public** knapsackMaxProfit(**int** itemName, **int** profit, **int** weight) {

**this**.**itemName** = itemName;

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

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

}

@Override

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

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

**else return** -1;

}

}

**class** maxProfitFindUsingMaxProfit

{

**public** ArrayList<knapsackMaxProfit> **arrayList**;

**public int maxCapacity**;

**public int**[] **itemName**;

**public int**[] **priceByItem**;

**public int index**;

**public double totalPrice**;

**public int**[] **weightByItem**;

**public** maxProfitFindUsingMaxProfit(ArrayList<knapsackMaxProfit> arrayList, **int** maxCapacity) {

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

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

**this**.**itemName** = **new int**[100];

**this**.**priceByItem** = **new int**[100];

**this**.**index** = 0;

**this**.**totalPrice** = 0;

**this**.**weightByItem** = **new int**[100];

}

**public void** giveItem()

{

**for** (knapsackMaxProfit give : **arrayList**)

{

calculateTotalAmount(give.**itemName**, give.**profit**, give.**weight**);

}

}

**public void** calculateTotalAmount(**int** item, **int** profit, **int** weight)

{

**if** (**maxCapacity** == 0) **return**;

**else if** (**maxCapacity**!=0)

{

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

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

{

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

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

**totalPrice** = **totalPrice** + fraction;

**itemName**[**index**] = item;

**priceByItem**[**index**] = profit;

**weightByItem**[**index**] = weight;

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

**maxCapacity** = 0;

}

**else if** (**maxCapacity**>0)

{

**totalPrice** = **totalPrice** + profit;

**itemName**[**index**] = item;

**priceByItem**[**index**] = profit;

**weightByItem**[**index**] = weight;

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

}

}

}

**public void** print()

{

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

{

System.***out***.println(**"Item No: "**+**itemName**[i]+**", Price is "**+**priceByItem**[i]+**", weight is "**+**weightByItem**[i]);

}

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

System.***out***.println(**"Total profit using max profit choose = "**+**totalPrice**);

}

}

**public class** knapsackUsingMaxProfitSolution {

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

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

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

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

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

{

arrayList.add(**new** knapsackMaxProfit((i+1),profit[i],weight[i]));

}

Collections.*sort*(arrayList);

**int** maxCapacity = 15;

maxProfitFindUsingMaxProfit object = **new** maxProfitFindUsingMaxProfit(arrayList,maxCapacity);

object.giveItem();

object.print();

}

}