**package** GreedyAlgorithm;

**import** java.util.ArrayList;

**import** java.util.Collections;

**class** ItemNameWeightPrice **implements** Comparable<ItemNameWeightPrice>

{

**public int itemName**;

**public int profit**;

**public int weight**;

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

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

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

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

}

@Override

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

**if** (**this**.**weight**>o.**weight**) **return** 1;

**else return** -1;

}

}

**class** takeItemWithMaxProfit

{

**public** ArrayList<ItemNameWeightPrice> **arrayList**;

**public double**[] **calculateProfit**;

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

**public int index**;

**public int capacity**;

**public double totalProfit**;

**public** takeItemWithMaxProfit(ArrayList<ItemNameWeightPrice> arrayList, **double**[] calculateProfit, **int** capacity) {

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

**this**.**calculateProfit** = calculateProfit;

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

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

**this**.**capacity** = capacity;

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

}

**public void** giveItem()

{

**for** (ItemNameWeightPrice give : **arrayList**)

{

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

}

}

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

{

**capacity** = **capacity** - weight;

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

**else if** (**capacity**<0)

{

**capacity** = **capacity** + weight;

**double** fractionProfit = Double.*valueOf*(profit) \* (Double.*valueOf*(**capacity**/ Double.*valueOf*(weight)));

**calculateProfit**[**index**] = fractionProfit;

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

**capacity** = 0;

**return**;

}**else**

{

**calculateProfit**[**index**] = profit;

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

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

}

}

**public void** print()

{

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

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

System.***out***.println(**"object: "**+**object**[i]+**" "**+**"profit = "**+**calculateProfit**[i]);

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

}

}

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

System.***out***.println(**"Total profit if we choose min weight = "**+**totalProfit**);

}

}

**public class** FractionalKnapsackMinWeight {

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

ArrayList<ItemNameWeightPrice> object = **new** ArrayList<>();

object.add(**new** ItemNameWeightPrice(1,5,1));

object.add(**new** ItemNameWeightPrice(2,10,3));

object.add(**new** ItemNameWeightPrice(3,15,5));

object.add(**new** ItemNameWeightPrice(4,7,4));

object.add(**new** ItemNameWeightPrice(5,8,1));

object.add(**new** ItemNameWeightPrice(6,9,3));

object.add(**new** ItemNameWeightPrice(7,4,2));

Collections.*sort*(object);

*/\* to calculate overall profit and max weight \*/*

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

**int** maxWeightCapacity = 15;

takeItemWithMaxProfit object1 = **new** takeItemWithMaxProfit(object,profit,maxWeightCapacity);

object1.giveItem();

object1.print();

}

}