**package** GreedyAlgorithmLab;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.HashMap;

**class** knapsackRatio **implements** Comparable<knapsackRatio>

{

**public int profit**;

**public int weight**;

**public double ratio**;

**public int itemName**;

**public** knapsackRatio(**int** profit, **int** weight, **double** ratio, **int** itemName) {

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

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

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

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

}

@Override

**public int** compareTo(knapsackRatio ratio) {

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

**else return** -1;

}

}

**class** findProfitUsingRatio

{

**public** HashMap<Double,Double> **hashMap**;

**public** ArrayList<knapsackRatio> **arrayList**;

**public int maxCapacity**;

**public double totalProfit**;

**public** findProfitUsingRatio(ArrayList<knapsackRatio> arrayList, **int** maxCapacity) {

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

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

**this**.**hashMap** = **new** HashMap<>();

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

}

**public void** giveItems()

{

**for** (knapsackRatio give : **arrayList**)

{

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

}

}

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

{

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

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

{

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

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

{

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

**double** fractionWeight = ((Double.*valueOf*(**maxCapacity**)/Double.*valueOf*(weight)));

**double** fraction = Double.*valueOf*(profit)\*(fractionWeight);

**totalProfit** = **totalProfit** + fraction;

**hashMap**.put(fraction,fractionWeight);

**maxCapacity** = 0;

}

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

{

**totalProfit** = **totalProfit** + profit;

**hashMap**.put(Double.*valueOf*(profit),Double.*valueOf*(weight));

}

}

}

**public void** print()

{

**for** (HashMap.Entry<Double,Double> print : **hashMap**.entrySet())

{

System.***out***.println(**"Profit is "**+print.getKey()+**", Weight is "**+print.getValue());

}

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

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

}

}

**public class** knapsackUsingRatio {

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

**int**[] profit = {60,100,120};

**int**[] weight = {10,20,30};

**double**[] ratio = **new double**[profit.**length**];

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

{

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

}

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

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

{

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

}

Collections.*sort*(arrayList);

**int** maxCapacity = 50;

findProfitUsingRatio knapsack = **new** findProfitUsingRatio(arrayList,maxCapacity);

knapsack.giveItems();

knapsack.print();

}

}