**package** AlgorithmDpPractise;

**import** java.util.ArrayList;

**import** java.util.Collections;

**class** sortingThroughWeight **implements** Comparable<sortingThroughWeight>

{

**public int weight**;

**public int profit**;

**public** sortingThroughWeight(**int** weight, **int** profit) {

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

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

}

@Override

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

**if** (**this**.**weight**>o.**weight**) **return** 1; *//ascending order sorting.......*

**else return** -1;

}

}

**class** knapsackUsingDp

{

**public int**[] **weight**;

**public int**[] **profit**;

**public int maxWeight**;

**public** ArrayList<sortingThroughWeight> **arrayList**;

**public int totalItems**;

**public int**[][] **solutions**;

**public int index**;

**public** knapsackUsingDp( ArrayList<sortingThroughWeight> arrayList,**int** maxWeight,**int** totalItems) {

**this**.**weight** = **new int**[totalItems];

**this**.**profit** = **new int**[totalItems];

**this**.**maxWeight**= maxWeight;

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

**this**.**totalItems** = totalItems;

**this**.**solutions** = **new int**[totalItems + 1][maxWeight+1];

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

}

**public void** initialize()

{

**for** (sortingThroughWeight giveData : **arrayList**)

{

**weight**[**index**] = giveData.**weight**;

**profit**[**index**] = giveData.**profit**;

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

}

}

**public void** findMaxProfit()

{

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

{

**solutions**[i][0] = 0;

}

**for** (**int** j=1;j<=**maxWeight**;j++)

{

**solutions**[0][j] = 0;

}

**for** (**int** i=1;i<=**weight**.**length**;i++)

{

**for** (**int** j=1;j<=**maxWeight**;j++)

{

**if** (**weight**[i-1]<=j)

{

**solutions**[i][j] = Math.*max*(**solutions**[i-1][j] , (**profit**[i-1] + **solutions**[i-1][j-**weight**[i-1]]));

}

**else**

{

**solutions**[i][j] = **solutions**[i-1][j];

}

}

}

System.***out***.println(**"The tabulation form is : "**);

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

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

{

**for** (**int** j=0;j<=**maxWeight**;j++)

{

System.***out***.print(**solutions**[i][j]+**" "**);

}

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

}

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

System.***out***.println(**"Max profit is = "**+**solutions**[**weight**.**length**][**maxWeight**]);

}

}

**public class** knapsackProblem {

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

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

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

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

**int** maxWeight = 50;

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

{

arrayList.add(**new** sortingThroughWeight(weight[i],profit[i]));

}

Collections.*sort*(arrayList);

knapsackUsingDp knapsack = **new** knapsackUsingDp(arrayList,maxWeight,weight.**length**);

knapsack.initialize();

knapsack.findMaxProfit();

}

}