# Ali Baba in Paradise

## Description

Ali Baba and his thieves enter a cave that contains a set of *n* **types of items**. Ali Baba and his thieves are free to choose **any type of item** and free to choose **any number of this type** as well. Each item type has both a *weight* and a *profit*. Their objective is to choose the set of items that maximizes the profit and their total weight doesn't exceed the maximum load of their camels. Note the following:

1- Each item should be **taken as a whole** (i.e. they can't take part of it)
2- They can take the same item **more than one time** (i.e. there're infinite instances of each item).

Given **N items** with the **weight & profit** of each, and the **Camels Maximum load**, find maximum profit that can be loaded on the Camels by the **OPTIMAL WAY**.

**Requirements:**

Implement TWO functions,

1. 1st function: return the max gained profit.
2. 2nd function: return the selected item(s) (if any) and the number of instances from each.

## Function:

### First Function:

```
  int SolveValue(int camelsLoad, int itemsCount, int[] weights, int[] profits)
<returns>Max total profit
```

### Second Function:

```
    Tuple<int, int>[] ConstructSolution(int camelsLoad, int itemsCount, int[]
                                  weights, int[] profits)
<returns>Tuple array of the selected items to get MAX profit (stored in
Tuple.Item1) together with the number of instances taken from each item (stored
in Tuple.Item2) OR NULL if no items can be selected
```

## Example

**Sample Input:**

N = 5, Load = 7

| Weight | Profit |
|---|---|
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |

**Sample Output:**

Max Profit = 5 (Can select one instance from the 2nd item and one instance for the 3rd item)

**Sample Input:**

N = 5, Load = 20

| Weight | Profit |
|---|---|
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |

**Sample Output:**

Max Profit = 16 (Can select one instance from the 1st item and three instance for the 5th item)

## C# Help

### TUPLE:

#### Creating a two-element tuple of integers

```
Tuple<int, int> t = new Tuple<int, int>(5, 7)
```

#### Accessing

`t.Item1` ➔ return 1st value (5 in the given example)

`t.Item2` ➔ return 2nd value (7 in the given example)

### ARRAYS:

#### Creating 1D array

```
int [] array = new int [size]
```

## Creating 2D array

```
int [,] array = new int [size1, size2]
```

## Length of 1D array

```
int arrayLength = my1DArray.Length
```

## Length of 2D array

```
int array1stDim = my2DArray.GetLength(0)

int array2ndDim = my2DArray.GetLength(1)
```

## Sorting single array

Sort the given array in ascending order

```
Array.Sort(items);
```

## Sorting parallel arrays

Sort the first array "master" and re-order the 2nd array "slave" according to this sorting

```
Array.Sort(master, slave);
```