Preparing present baskets

A loving daughter decided to prepare a precious gift for her parents. She bought two baskets and decided to fill each basket with the most expensive fruits from the supermarket. Each basket can be filled with maximum weight.

In the supermarket; there're 'N' fruits. For each one, the supermarket displays the full remaining weights 'weights[i]' with their total costs 'costs[i]'. Design an **efficient algorithm** so that the daughter can fill her baskets with the most expensive collection of fruits.

Function to Implement

public static `double` **PreparePresentBaskets**(`int` W1, `int` W2, `KeyValuePair`<int, int>[] items)

`PresentBaskets.cs includes this method.`

- W1: weight of 1st basket
- W2: weight of 2nd basket
- Items: Pair of weight (Key) & cost (Value) of each item

<returns> max total cost to fill two baskets

Example

Sample Input:

$W_1 = 10$, $W_2 = 80$

Item	Weight	Cost
1	30	10$
2	10	50$
3	20	20$
4	50	150$
5	40	80$
6	25	200$

Sample Output:

Max Cost = 400$ *(from items 2,4,6)*

C# Help

Creating 1D array

```
int [] array1D = new int [size]
```

Creating 2D array

```
int [,] array2D = new int [size1, size2]
```

Getting the size of 1D array

```
int size = array1D.GetLength(0);
```

Getting the size of 2D array

```
int size1 = array2D.GetLength(0);

int size2 = array2D.GetLength(1);
```

Sorting single array

Sort the given array "items" 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);
```