Bread Transportation

Description

City X consists of one street, and every inhabitant in the city is a bread salesman. Simple enough: everyone buys bread from other inhabitants of the city. Every day each inhabitant decides how much bread he wants to buy or sell. Interestingly, demand and supply are always the same, so that each inhabitant gets what he wants.

There is one problem, however: Transporting bread from one house to another results in work, since all bread is equally good, the inhabitants of City X don't care which persons they are doing trade with, they are only interested in selling or buying a specific amount of bread. They are cleaver enough to figure out a way of trading so that the overall amount of work needed for transports is minimized.

In this problem you are asked to reconstruct the trading during one day in City X. For simplicity we will assume that the houses are built along a straight line with equal distance between adjacent houses. Transporting one loaf of bread from one house to an adjacent house result in one unit of work.

Input: **Already Implemented**

It takes an array of N integers (each array index is House number and each Value corresponds to amount of bread needed or owned by the house) e.g. array {5 ,-4 ,1 , -3,1} it means house 1 has 5 loafs of bread to sell , house 2 wants to buy 4 loafs of bread , house 3 wants to sell 1 loaf of bread , house 4 wants to buy 3 loafs of bread , house 5 owns 1 loaf of bread. The total sum of array values is 0

Output: **Already Implemented**

Output should be the amount of minimum work units needed to fulfil Bread demands from one house to another

E.g. `5 -4 1 -3 1 input will result in 9 output (4 breads and 1 work unit from house 1 to 2 is 4*1 = 4 , and 3 work units and 1 bread from house 1 to 4 is 1*3=3 , 1 work unit and 1 bread from 3 to 4 is 1*1=1 , 1 from 5 to 4)`

Complexity

complexity of your algorithm should be **less than $O(N^2)$**

Function: **Implement it!**

```
public static Int64 RequiredFunction(int N ,int[] DemandPerHouse)
```

you will find function in file PROBLEM_CLASS.cs

Example

#	Input Array	Output
1	5 5 -4 1 -3 1	9
2	6 -1000 -1000 -1000 1000 1000 1000	9000
3	9 **2 4 -9 -8 -4 -3 7 6 5**	71
4	10 1 2 3 4 5 -5 -4 -3 -2 -1	55

C# Help

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);
```

Creating 1D array

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

Creating 2D array

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

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);
```