

Ali Baba assembling a team

Ali Baba heard about a new cave to be robbed, so he needs to assemble his team of thieves to do the job, some thieves need to be paid to go with him, some might pay money just to be with Ali Baba, others don't get paid or pay.

Anyway, the thieves sit in a long line and Ali Baba has to choose a group of them and this group should cost him as minimum as possible, where thieves who get paid cost him +ve LE, those who pay him to come cost him -ve LE, and the others cost him nothing 0 LE

But, **he can't select from them except 1 group of consecutive thieves**, like he can say I want all the thieves between the thief number 3 and the thief number 8, so his group would be the thieves 3, 4, 5, 6, 7 and 8, adding the costs of each one will lead to the total cost of this group.

So, the problem is.. given the number of thieves sitting in front of Ali Baba (**N**), followed by the cost of each thief, Design the **most efficient algorithm** to find the minimum cost for any team that can be assembled (its ok to have a team of 1 thief)

NOTE: Ali Baba MUST form a team (even of one thief)

Function to Implement

```
static public long AssembleTeam(int N, short[] array)
```

AliBabaAssembleTeam.cs includes this method.

Example

Sample Input:

N = 10 ; Costs = 0, 5, -3, 4, -3, -6, -1, 9, -16, 30

Sample Output:

-17

(Note: The group Ali Baba will choose is from number 4 to number 8, and they will cost him -17 LE)

0, 5, -3, 4, -3, -6, -1, 9, -16, 30

-3 + -6 + -1 + 9 + -16 = -17

Sample Input:

N = 4 ; Costs = 11, 5, 3, 4

Sample Output:

3

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);
```