**5 kyu**

**Compute the Largest Sum of all Contiguous Subsequences**

20493% of 6858 of218[LesRamer](https://www.codewars.com/users/LesRamer)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/compute-the-largest-sum-of-all-contiguous-subsequences/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/compute-the-largest-sum-of-all-contiguous-subsequences/solutions/csharp)

[Discourse (11)](https://www.codewars.com/kata/compute-the-largest-sum-of-all-contiguous-subsequences/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Given an array of numbers, calculate the largest sum of all possible blocks of consecutive elements within the array. The numbers will be a mix of positive and negative values. If all numbers of the sequence are nonnegative, the answer will be the sum of the entire array. If all numbers in the array are negative, your algorithm should return zero. Similarly, an empty array should result in a zero return from your algorithm.

largestSum([-1,-2,-3]) == 0

largestSum([]) == 0

largestSum([1,2,3]) == 6

Easy, right? This becomes a lot more interesting with a mix of positive and negative numbers:

largestSum([31,-41,59,26,-53,58,97,-93,-23,84]) == 187

The largest sum comes from elements in positions 3 through 7: 59+26+(-53)+58+97 == 187

Once your algorithm works with these, the test-cases will try your submission with increasingly larger random problem sizes.

<https://www.codewars.com/kata/compute-the-largest-sum-of-all-contiguous-subsequences/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int LargestSum(int[] arr)

{

// your code to return the largest sum of any subsequence of arr

if (arr.Length == 0) return 0;

int max\_sum = arr[0];

int max = arr[0];

for(int i =1; i<arr.Length; i++)

{

max = Math.Max(arr[i], max + arr[i]);

max\_sum = Math.Max(max, max\_sum);

}

if (max\_sum < 0) return 0;

return max\_sum;

}

static void Main(string[] args)

{

Console.ReadLine();

}

}

}