Given a list of n positive integers, count the numbers of *k-Sub sequences* in it. A *k-Sub sequence* is defined as follows:

* it is a sequence of consecutive elements ofn;
* the sum of its elements is evenly divisible byk.

**Example**

For k = 3 and nums = [1, 2, 3, 4, 1], the output should be  
ksubsequences(k, nums) = 4.

There're 4 *3-Sub sequences* in nums:

* [1, 2]
* [1, 2, 3]
* [2, 3, 4]
* [3]

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer k**

*Constraints:*  
1 ≤ k ≤ 100.

* **[input] array.integer nums**

*Constraints:*  
1 ≤ nums.length ≤ 5000,  
1 ≤ nums[i] ≤ 10000.

* **[output] integer**

The number of *k-Sub sequences* in nums.

<https://codefights.com/challenge/DBDMfEj46wcyZovys/main>

static int ksubsequences(int k, int[] nums)

{

int cont = 0;

for (int i = 0; i < nums.Length; i++)

{

int sum = 0;

for (int j = i; j < nums.Length; j++)

{

sum += nums[j];

if (sum % k == 0)

{

cont++;

}

}

}

return cont;

}