The problem asks if there exists non negative integer solutions to the following equation:

A1*x1 + A2*x2 + A3*x3 + ... AN*xn = S

X's are given, and we need non negative integer solutions for A's.

Let's call a number representable if we can find a non negative integer solutions for it.

Probably the first idea is the similarity of this problem and the linear Diophantine equation (https://en.wikipedia.org/wiki/Diophantine_equation),but the condition on the coefficients being non negative makes it really hard to solve (if not impossible) in reasonable time. Another thing to look at is the constraints are actually small for a number theory problem (n <= 5000). I dropped these because I think it's too easy to get stuck trying to force a complete number theory/math solution.

if S is zero, we just output yes, set all A's to zero otherwise then wlog there exists a latest xi we add. Choose any xi.

So we can write S as (y) + k*xi where Y is representable from xi's by some chosen coefficients. (y >= 0)

Now let's do some simple math :)

S = Y + k*(xi)

Let's apply modulo xi to the whole equation,we will get rid of k*xi as multiple of xi modulo xi is zero.

S % xi = Y % xi

So now, we need to find a representable number from the xi's that is equal to S % xi, if that's not possible then by contradiction S is not representable.

Let's say we found a way to do so, a solution would only exist if the sum of that set is >= S. (Because once that is done, we can keep adding multiples of the original xi till we reach it).

So for each modulo from 1 to xi, we need to find the smallest sum to reach it from a given N numbers.

This is a hidden Dijkstra problem. Let the nodes be numbered from 0 to (xi - 1), and an edge exists between two nodes a and b with cost arr[j] if (a + arr[j]) % xi is equal to j.

We can run dijikstra, and find for each modulo (0,(xi - 1)) the minimum cost needed to reach it.

For any query we just need to check if S >= distance[S % xi]. (Make sure initially the dijikstra distance array is set to a large value).

https://github.com/MedoN11/CompetitiveProgramming/blob/master/POI/10SUMS.cpp