Find anagrams in linked list

Given a linked list of characters and a string S.Return all the anagrams of the string present in the given linked list.In case of overlapping anagrams choose the first anagram from left.

Example 1:

```
Input: a -> b -> c -> a -> d -> b -> c -> a
S = bac
Output: [a -> b -> c, b -> c -> a]
Explanation: In the given linked list,
there are three anagrams:
1. a -> b -> c -> a -> d -> b -> c -> a
2. a -> b -> c -> a -> d -> b -> c -> a
3. a -> b -> c -> a -> d -> b -> c -> a
But in 1 and 2, a -> b -> c and b -> c-> a
are ovelapping.So we take a -> b -> c as it
comes first from left.So the output is:
[a->b->c,b->c->a]
```

Example 2:

```
Input: a -> b -> d -> c -> a
S = bac
Output: -1
Explanation: There is no anagrams, so output is -1
```

Your Task:

You don't need to read input or print anything. Your task is to complete the function **findAnagrams()** which takes head node of the linked list and a string S as input parameters and returns an array of linked list. If there is no anagram in the linked list,return -1.

Expected Time Complexity: O(N), where N is length of LinkedList
Expected Auxiliary Space: O(1)

Constraints:
$1 <= N <= 10^5$
$1 <= |S| <= 10^5$