A Needle in the Haystack

Write a program that finds all occurences of a given pattern in a given input string. This is often referred to as finding a *needle* in a *haystack*.

The program has to detect **all** occurences of the needle in the haystack. It should take the needle and the haystack as input, and output the positions of each occurence, as shown below. The suggested implementation is the KMP algorithm, but this is not a requirement. However, a naive approach will probably exceed the time limit, whereas other algorithms are more complicated... The choice is yours.

Input

The input consists of a number of test cases. Each test case is composed of three lines, containing:

- the length of the needle,
- the needle itself,
- the haystack.

The length of the needle is only limited by the memory available to your program, so do not make any assumptions - instead, read the length and allocate memory as needed. The haystack is **not** limited in size, which implies that your program should not read the whole haystack at once. The KMP algorithm is stream-based, i.e. it processes the haystack character by character, so this is not a problem.

The test cases come one after another, each occupying three lines, with no additional space or line breaks in between.

Output

For each test case your program should output all positions of the needle's occurences within the haystack. If a match is found, the output should contain the position of the first character of the match. Characters in the haystack are numbered starting with zero.

For a given test case, the positions output should be sorted in ascending order, and each of these should be printed in a separate line. For two different test cases, the positions should be separated by an empty line.

Example

Sample input:

```
2
na
bananobano
6
foobar
foo
9
foobarfoo
barfoobarfoobarfoobarfoo
```

Sample output:

```
2
4

3
9
15
21
```

Note the double empty line in the output, which means that no match was found for the second test case.

Warning: large Input/Output data, be careful with certain languages