Problem Statement **Suggest Edit**

Given two strings ‘STR’ and ‘PTR’. Find all the starting indices of ‘PTR’ anagram substring in ‘STR’. Two strings are anagram if and only if one string can be converted into another string by rearranging the character.

For example, ‘ABCD’ and ‘ACBD’ are two anagram strings because ‘ACBD’ can be converted into ‘ABCD’ by rearranging the ‘B’ and ‘C’. ’ABA’ and ‘ABB’ are not anagram because we can’t convert ‘ABA’ to ‘ABB’ by rearranging the characters of particular strings.

‘ABACD’ and ‘CABAD’ are anagram because ‘ABACD’ can be converted into ‘CABAD’ by rearranging the first ‘A’ with ‘C’ and second ‘A’ with ‘B’.

Note:

Strings ‘STR’ and ‘PTR’ consist only of English uppercases.

Length of string ‘STR’ will always be greater than or equal to the length of string ‘PTR’.

The index is ‘0’ based.

In case, there is no anagram substring then return an empty sequence.

Explanation:

For example, the given ‘STR’ is ‘BACDGABCD’ and ‘PTR’ is ‘ABCD’. Indices are given

0-3 in ‘STR’ index 0,1,2,3 are ‘BACD’ and it is an anagram with ‘ABCD’
1-4 in ‘STR’ index 1,2,3,4 are ‘ACDG’ and it is not anagram with ‘ABCD’
2-5 in ‘STR’ index 2,3,4,5 are ‘CDGA’ and it is not anagram with ‘ABCD’
3-6 in ‘STR’ index 3,4,5,6 are ‘DGAB’ and it is not anagram with ‘ABCD’
4-7 in ‘STR’ index 4,5,6,7 are ‘GABC’ and it is not anagram with ‘ABCD’
5-8 in ‘STR’ index 5,6,7,8 are ‘ABCD’ and it is an anagram with ‘ABCD’

Hence there are 2 starting indices of substrings in the string ‘STR’ that are anagram with given ‘PTR’ which are index 0 and 5.

Input Format:

The first line of input contains an integer ‘T’ denoting the number of test cases.

The next ‘3*T’ lines represent the ‘T’ test cases.

The first line of each test case contains two integers ‘N’ and ‘M’. Where ‘N’ denotes the number of characters in ‘STR’ and ‘M’ denotes the number of characters in ‘PTR’.

The second line of each test case contains the string ‘STR’ on a separate line .

The third line of each test case contains the string ‘PTR’ on a separate line.

Output Format

For each test case, print a sequence of all the starting indices of the anagram substrings present in the given word/string ‘STR’.

The output of each test case will be printed in a separate line.

Note:

You do not need to print anything; it has already been taken care of. Just implement the given function.

Constraints:

```
1 <= 'T' <= 50
1 <= 'N', 'M' <= 10 ^ 4
```

Where ‘T’ is the total number of test cases, ‘N’ denotes the number of characters in the first given string ‘STR’ and ‘M’ denotes the number of characters in the second given string ‘PTR’. Strings ‘STR’ and ‘PTR’ only consist of English uppercase alphabets.

Time limit: 1 second.

Sample Input 1:

```
2
10 3
CBAEBABACD
ABC
5 2
ABADE
BA
```

Sample Output 1:

```
0 6
0 1
```

Explanation Of Sample Input 1::

Test Case 1:

‘str’ is ‘CBAEBABACD’ and ‘ptr’ is ‘ABC’

0-2 in ‘str’ index 0,1,2 are ‘CBA’ and it is an anagram with ‘ABC’
1-3 in ‘str’ index 1,2,3 are ‘BAE’ and it is not anagram with ‘ABC’
2-4 in ‘str’ index 2,3,4 are ‘AEB’ and it is not anagram with ‘ABC’
3-5 in ‘str’ index 3,4,5 are ‘EBA’ and it is not anagram with ‘ABC’
4-6 in ‘str’ index 4,5,6 are ‘BAB’ and it is not anagram with ‘ABC’
5-7 in ‘str’ index 5,6,7 are ‘ABA’ and it is not anagram with ‘ABC’
6-8 in ‘str’ index 6,7,8 are ‘BAC’ and it is an anagram with ‘ABC’
7-9 in ‘str’ index 7,8,9 are ‘ACD’ and it is not anagram with ‘ABC’

Hence there are only two anagram substrings in the given string ‘str’ that are anagram with given string ‘ptr’ which are ‘CBA’ and ‘BAC’ and starting indices of respective anagram substrings are 0 and 6.

Test case 2:

‘str’ is ‘ABADE’ and ‘ptr’ is ‘BA’.

In the given string ‘ABADE’ the substring of length 2 starting with index 0 is ‘AB’ which is an anagram with the string ‘BA’ and a substring of length 2 starting with index 1 is ‘BA’ which is also an anagram with the string ‘BA’. Because 0 and 1 are starting indices of the substrings, we return 0 and 1.

Sample Input 2:

```
2
10 4
BACDGABCDA
ABCD
7 1
ABABABA
A
```

Sample Output 2:

```
0 5 6
0 2 4 6
```