FINDING A MOTIF IN DNA

Problem

Given two strings s and t, t is a **substring** of s if t is contained as a contiguous collection of symbols in s (as a result, t must be no longer than s).

The **position** of a symbol in a string is the total number of symbols found to its left, including itself (e.g., the positions of all occurrences of 'U' in "AUGCUUCAGAAAGGUCUUACG" are 2, 5, 6, 15, 17, and 18). The symbol at position i of s is denoted by s[i].

A substring of s can be represented as s[j:k], where j and k represent the starting and ending positions of the substring in s; for example, if s = "AUGCUUCAGAAAGGUCUUACG", then s[2:5] = "UGCU".

The **location** of a substring s[j:k] is its beginning position j; note that t will have multiple locations in s if it occurs more than once as a substring of s (see the Sample below).

Given:

Two DNA strings s and t (each of length at most 1 kbp).

Return:

All locations of t as a substring of s.

Sample Dataset

```
GATATATGCATATACTT
ATAT
```

Sample Output

```
2 4 10
```

NOTE

Different programming languages use different notations for positions of symbols in strings. Above, we use **1-based numbering**, as opposed to **0-based numbering**, which is used in Python. For s = "AUGCUUCAGAAAGGUCUUACG", 1-based numbering would state that s[1] = 'A' is the first symbol of the string, whereas this symbol is represented by s[0]in 0-based numbering. The idea of 0-based numbering propagates to substring indexing, so that s[2:5] becomes "GCUU" instead of "UGCU".

Note that in some programming languages, such as Python, s[j:k] returns only fragment from index j up to but *not* including index k, so that s[2:5] actually becomes "UGC", not "UGCU".