1H Find All Approximate Occurrences of a Pattern in a String

Approximate Pattern Matching Problem
Find all approximate occurrences of a pattern in a string.

Input: DNA strings *Pattern* and *Text* along with an integer d.
Output: All starting positions where *Pattern* appears as a substring of *Text* with at most d mismatches.

```
CGACTAGTTT
CGACGA
0  3
```

Formatting

Input: DNA strings *Pattern* and *Text* along with an integer d.
Output: A space-separated list of integers containing all starting positions where *Pattern* appears as a substring of *Text* with at most d mismatches.

Constraints

- The length of *Pattern* will be between 1 and 10^2.
- The length of *Text* will be between 1 and 10^5.
- The integer d will be between 1 and 10^1.
- *Pattern* and *Text* will be DNA strings.

Test Cases

Case 1

Description: The sample dataset is not actually run on your code.

Input:

```
ATTCTGGA
CGCCCGAATCCAGAACGCATTCCCATATTTCGGGACCACTGGCCTCCACGGTACGGACGTCAATCAAAT
3
```

Output:

```
6 7 26 27
```

Case 2

Description: This dataset checks if you are only counting instances where the number of mismatches is exactly equal to d (i.e. ignoring instances where mismatch < d).

Input:

```
AAA
TTTTTTAAATTTTAAATTTTTT
2
```

Output:

```
4 5 6 7 8 11 12 13 14 15
```

Case 3

Description: This dataset checks if your code has an off-by-one error at the beginning of *Text* (i.e. your code is not checking the the left-most substring of *Text*).

Input:

```
GAGCGCTGG
GAGCGCTGGGTTAACTCGCTACTTCCCGACGAGCGCTGTGGCGCAAATTGGCGATGAAACTGCAGAGAGAACTG...
...GTCATCCAACTGAATTCTCCCCGCTATCGCATTTTGATGCGCGCCGCGTCGATT
2
```

Output:

```
0 30 66
```

Case 4

Description: This dataset checks if your code has an off-by-one error at the end of *Text* (i.e. your code is not checking the the right-most substring of *Text*).

Input:

```
AATCCTTTCA
CCAAATCCCCTCATGGCATGCATTCCCGCAGTATTTAATCCTTTCATTCTGCATATAAGTAGTGAAGGT...
...ATAGAAACCCGTTCAAGCCCGCAGCGGTAAAACCGAGAACCATGATGAATGCACGGCGATTGCGCC...
...ATAATCCAAACA
3
```

Output:

```
3 36 74 137
```

Case 5

Description: This dataset checks if your code is correctly accounting for overlapping instances of *Pattern* in *Text*.

Input:

```
CCGTCATCC
CCGTCATCCGTCATCCTCGCCACGTTGGCATGCATTCCGTCATCCCGTCAGGCATACTTCTGCATATAA...
...GTACAAACATCCGTCATGTCAAAGGGAGCCCGCAGCGGTAAAACCGAGAACCATGATGAATGCACG...
...GCGATTGC
3
```

Output:

```
0 7 36 44 48 72 79 112
```

Case 6

Description: This dataset checks if you are only counting instances of *Pattern* with less than *d* mismatches (as opposed to instances of *Pattern* with less than or equal to *d* mismatches).

Input:

```
TTT
AAAAAA
3
```

Output:

```
0 1 2 3
```

Case 7

Description: This dataset checks if your code works with input where $d = 0$ (i.e. only perfect matches are allowed).

Input:

```
CCA
CCACCT
0
```

Output:

```
0
```

Case 8

Description: A larger dataset of the same size as that provided by the randomized autograder. Check input/output folders for this dataset.