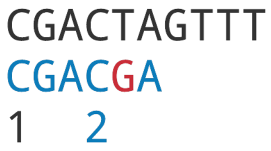
1O Implement ApproximatePatternCount

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

Input: Strings *Pattern* and *Text* as well as an integer d.
Output: $\textsc{Count}_d(\textit{Text}, \textit{Pattern})$.



Formatting

Input: A DNA string *Pattern* followed by a DNA string *Text*, followed by an integer d.
Output: A single integer $\textsc{Count}_d(\textit{Text}, \textit{Pattern})$.

Constraints

- The length of *Pattern* will be between 1 and 10^1.
- The length of *Text* will be between 1 and 10^3
- The integer d will be between 1 and 10^1.
- Both *Pattern Text* and will be DNA strings.

Test Cases

Case 1

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

Input:

```
ATA
CGATCGAGTACCATAAG
1
```

Output:

```
2 7 12
```

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:

```
GAGC
GAGCGCTACTTCCCGACGAGCGCTTGA
2
```

Output:

```
0 2 14 17 19
```

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:

```
AATC
CGATGCATTAAATCC
2
```

Output:

```
1 2 5 6 9 10 11
```

Case 5

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

Input:

```
CCC
ACCCGCCCTCCCGGC
1
```

Output:

```
0 1 2 3 4 5 6 7 8 9 10
```

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.