Frequent Words Problem

Input: A string *Text* and an integer *k*

Output: All most frequent *k*-mers in *Text*

Pseudocode

```
FrequentWordsProblem(Text,k)
    counts ← 0 for all possible kmers
    for i ← 0 to |Text| - |Pattern|
        kmer ← Text(i,|Pattern|)
        counts(kmer) ← counts(kmer) + 1
            count ← count + 1
    return all kmers where counts(kmer) = max(counts)
```

SAMPLE DATASET:

<u>Input</u>:

```
ACGTTGCATGTCGCATGATGCATGAGAGCT

4
```

<u>Output</u>:

```
CATG GCAT
```

The sample dataset is not actually run on your code.

TEST DATASET 1:

Input:

```
TGGTAGCGACGTTGGTCCCGCCGCTTGAGAATCTGGATGAACATAAGCTCCCACTTGGCTTATT
CAGAGAACTGGTCAACACTTGTCTCTCCCAGCCAGGTCTGACCACCGGGCAACTTTTAGAGCAC
TATCGTGGTACAAATAATGCTGCCAC

3
```

Output:

```
TGG
```

This dataset just checks if you're counting the first kmer in Text (TGG in this example). If you do not count the first kmer (TGG), you will get the following "most frequent" kmers in addition to TGG:

ACT CAC CCA CTT GGT

TEST DATASET 2:

Input:

```
CAGTGGCAGATGACATTTTGCTGGTCGACTGGTTACAACAACGCCTGGGGCTTTTGAGCAACGA
GACTTTTCAATGTTGCACCGTTTGCTGCATGATATTGAAAACAATATCACCAAATAAATAACGC
CTTAGTAAGTAGCTTTT

4
```

Output:

```
TTTT
```

This dataset just checks if you're counting the last kmer in Text (TTTT in this example). If you do not count the last kmer (TTTT), you will get the following "most frequent" kmers in addition to TTTT:

AACG   AATA   ACAA   CAAC   CTGG   CTTT   TTGC   TTTG

TEST DATASET 3:

Input:

```
ATACAATTACAGTCTGGAACCGGATGAACTGGCCGCAGGTTAACAACAGAGTTGCCAGGCACTG
CCGCTGACCAGCAACAACAACAATGACTTTGACGCGAAGGGGATGGCATGAGCGAACTGATCGT
CAGCCGTCAGCAACGAGTATTGTTGCTGACCCTTAACAATCCCGCCGCACGTAATGCGCTAACT
AATGCCCTGCTG

5
```

Output:

```
AACAA
```

This dataset checks if your code correctly handles cases where there are overlapping occurrences of Pattern throughout Text. For example, AACAACAA contains two occurrences of AACAA (**AACAA**CAA and AAC**AACAA**), so if your code counts AACAACAA as one occurrence of AACAA, your code will fail on this test case.

TEST DATASET 4:

<u>Input</u>:

```
CCAGCGGGGGTTGATGCTCTGGGGGTCACAAGATTGCATTTTTATGGGGTTGCAAAAATGTTTT
TTACGGCAGATTCATTTAAAATGCCCACTGGCTGGAGACATAGCCCGGATGCGCGTCTTTTACA
ACGTATTGCGGGGTAAAATCGTAGATGTTTTAAAATAGGCGTAAC

5
```

<u>Output</u>:

```
AAAAT GGGGT TTTTA
```

This test dataset checks if your code correctly handles ties (i.e. your code actually outputs ALL "most frequent" kmers, and not just a single "most frequent" kmer). For example, in the string "ATATA", there are two "most-frequent" kmers: "AT" and "TA". "AT" occurs twice (**ATAT**A), and "TA" occurs twice (A**TATA**), so both of these should be outputted (separated by a space character).