Clump Finding Problem

Input: A string *Genome*, and integers *k*, *L*, and *t*

Output: All distinct *k*-mers forming (*L*, *t*)-clumps in *Genome*

Pseudocode

```
ClumpFinding(Genome,k,L,t)
    for i ← 0 to |Genome| - L
        count ← 0 for all kmers in Genome(i,L)
        for j ← 0 to L - k
            kmer = Genome(i+j,L)
            count(kmer) = count(kmer) + 1
        for all kmers in count
            if count(kmer) >= t and kmer has not been outputted
                output kmer
```

SAMPLE DATASET:

Input:

```
CGGACTCGACAGATGTGAAGAACGACAATGTGAAGACTCGACACGACAGAGTGAAGAGAAGAGG
AAACATTGTAA

5 50 4
```

Output:

```
CGACA GAAGA
```

The sample dataset is not actually run on your code.

TEST DATASET 1:

Input:

```
AAAACGTCGAAAAA

2 4 2
```

Output:

```
AA
```

This dataset makes sure that your code only counts kmers that fall COMPLETELY within a given L-window. For example, take the 4-window starting at index 4 (AAAA**CGTC**GAAAAA). One might think that the 2-mer "CG" occurs twice in this window since the first letter of the second occurrence happens at the very end of the window. However, since the second occurrence of "CG" does not fall entirely in this 4-window, it does not count. Thus, the only result is "AA".

TEST DATASET 2:

Input:

```
ACGTACGT

1 5 2
```

Output:

```
A C G T
```

This dataset checks if your code has an off-by-one error when checking kmers within an L-window. Notice that, for each 1-mer (A, C, G, and T), there are 3 nucleotides between the first and second occurrence. In other words, each nucleotide occurs twice in a specific 5-window: once at the beginning of the 5-window, and once at the end: **ACGTACGT, ACGTACGT, ACGTACGT, and ACGTACGT.**

TEST DATASET 3:

Input:

```
CCACGCGGTGTACGCTGCAAAAAGCCTTGCTGAATCAAATAAGGTTCCAGCACATCCTCAATGG
TTTCACGTTCTTCGCCAATGGCTGCCGCCAGGTTATCCAGACCTACAGGTCCACCAAAGAACTT
ATCGATTACCGCCAGCAACAATTTGCGGTCCATATAATCGAAACCTTCAGCATCGACATTCAAC
ATATCCAGCG

3 25 3
```

Output:

```
AAA CAG CAT CCA GCC TTC
```

This dataset checks if your code is correctly handling overlapping kmers. For example, "ATA" forms a (5, 2)-clump in CCCATATACCC (CCC**ATA**<u>**TA**</u>CCC and CCC**AT**<u>**ATA**</u>**CCC).**