Our goal is to modify our previous algorithm to find *DnaA* boxes by identifying frequent *k*-mers, possibly mismatches. Given strings *Text* and *Pattern* as well as an integer *d*, we define $Count_d(Text, Pattern)$ as the number of occurrences of *Pattern* in *Text* with at most *d* mismatches. example, $Count_1$(AACAAGCTGATAAACATTTAAAGAG, AAAAA) = 4 because AAAAA appears four times in this st with at most one mismatch: AACAA, ATAAA, AAACA, and AAAGA. Note that two of these occurrences overlap.

A **most frequent *k*-mer with up to *d* mismatches** in *Text* is simply string *Pattern* maximizing $Count_d(Text, Pattern)$ among all *k*-mers. Note that *Pattern* does not need to actu appear as a substring of *Text*; for example, as we saw above, AAAAA is the most frequent 5-mer with 1 mism inAACAAGCTGATAAACATTTAAAGAG, even though it does not appear in this string. Keep this in mind while sol the following problem:

Frequent Words with Mismatches Problem: *Find the most frequent k-mers with mismatches in a string*.

Input: A string *Text* as well as integers *k* and *d*. (You may assume $k \leq 12$ and $d \leq 3$.)

Output: All most frequent *k*-mers with up to *d* mismatches in *Text*.

CODE CHALLENGE: Solve the Frequent Words with Mismatches Problem.

Sample Input:

```
ACGTTGCATGTCGCATGATGCATGAGAGCT 4 1
```

Sample Output:

```
GATG ATGC ATGT
```