Frequent Words with Mismatches Problem

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*.

SAMPLE DATASET:

Input:

```
ACGTTGCATGTCGCATGATGCATGAGAGCT
4 1
```

Output:

```
ATGC ATGT GATG
```

The sample dataset is not actually run on your code.

TEST DATASET 1:

Input:

```
AAAAAAAAAA
2 1
```

Output:

```
AA AC AG CA AT GA TA
```

This dataset checks that your code includes *k*-mers that do not actually appear in *Text*. Notice here that, although none of the output *k*-mers except for AA actually appear in *Text*, they are all valid because they appear in *Text* with up to 1 mismatch (i.e. 0 or 1 mismatch).

TEST DATASET 2:

Input:

```
AGTCAGTC
4 2
```

Output:

```
TCTC CGGC AAGC TGTG GGCC AGGT ATCC ACTG ACAC AGAG ATTA TGAC AATT
CGTT GTTC GGTA AGCA CATC
```

This dataset makes sure that your code is not accidentally swapping *k* and *d*.

TEST DATASET 3:

Input:

```
AATTAATTGGTAGGTAGGTA
4 0
```

Output:

```
GGTA
```

This dataset makes sure you are not finding patterns in the Reverse Complement of *Text* (that is the next problem, "Frequent Words with Mismatches and Reverse Complements Problem").

TEST DATASET 4:

Input:

```
ATA
3 1
```

Output:

```
GTA ACA AAA ATC ATA AGA ATT CTA TTA ATG
```

This dataset first checks that *k*-mers with exactly *d* mismatches are being found. Then, it checks that *k*-mers with less than *d* mismatches are being allowed (i.e. you are not only allowing *k*-mers with exactly *d* mismatches). Next, it checks that you are not returning too few *k*-mers. Last, it checks that you are not returning too many *k*-mers.

TEST DATASET 5:

Input:

```
AAT
3 0
```

Output:

```
AAT
```

This dataset checks that your code is not looking for *k*-mers in the Reverse Complement of *Text*.

TEST DATASET 6:

Input:

```
TAGCG
2 1
```

Output:

```
GG TG
```

This dataset checks that your code correctly delimiting your output (i.e. using spaces) and verifies that your *k*-mers are actually of length *k*.