Frequent Words with Mismatches and Reverse Complements Problem

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

**Output:** All $k$-mers *Pattern* maximizing the sum $\text{Count}_d(\text{Text}, \text{Pattern}) + \text{Count}_d(\text{Text}, \text{Pattern})$ over all possible $k$-mers.
SAMPLE DATASET:

Input:
ACGTTGCATGTCGATGATGAGAGCT
4 1

Output:
ACAT ATGT

The sample dataset is not actually run on your code.
**TEST DATASET 1:**

**Input:**
AAAAAAAAAA
2 1

**Output:**
AT TA

This dataset checks that your code includes $k$-mers that do not actually appear in *Text*. Notice here that, although AT nor TA actually appear in *Text*, they are 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:
AATT GGCC

This dataset makes sure that your code is not accidentally swapping $k$ and $d$. 
TEST DATASET 3:

Input:
AATTAATTGGTAGGTAGGTA
4 0

Output:
AATT

This dataset makes sure you are finding k-mers in both Text and the Reverse Complement of Text.
TEST DATASET 4:
 Input:
ATA
3 1
 Output:
AAA AAT ACA AGA ATA ATC ATG ATT CAT CTA GAT GTA TAA TAC TAG TAT TCT TGT TTA TTT

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 ATT

This dataset checks that your code is looking at BOTH Text and its Reverse Complement (i.e. not just looking at Text, and not just looking at the Reverse Complement of Text, but looking at both).
TEST DATASET 6:

Input:
TAGCG
2 1

Output:
CA CC 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 $k$. 