Goal

The goal of this task is to write a function grade() that grades a student's answer and finds certain common mistakes. In particular, the program should figure out (1) if there is a typo in the student's answer, (2) if a word is missing or (3) if a word is wrong. If the program finds a common mistake it should highlight it.

Specification

Here is the specification in Python (you do not have to use python). Make sure to make the function available from the command line.

```
def grade(correct_answer, student_answer):
"""
    - correct_answer: a unicode string, the correct answer
    - student_answer: a unicode string, what the student typed

    returns a tuple (correct, blame, highlights)
        - correct:      a boolean, True if and only if the student_answer
                        should be considered a correct answer
        - blame:        one out of {None, "typo", "missing", "wrong_word"}
                        depending on the cause of the mistake, if it can
                        be detected
        - highlights:   a list of tuples, where each tuple is of type
                        ((c1, c2), ((s1, s2)) and c1/s1 is the index of the
                        first character of a blamed word in
                        the correct/student's answer and c2/s2 is the index
                        of the last character of that same blamed word

    Examples:

    >>> grade("house", "house")
    (True, None, [])

    >>> grade("house", "house.")
    (True, None, [])

    >>> grade("A house", "a house.")
    (True, None, [])

    >>> grade("house", "hhouse")
    (True, "typo", [((0,5), (0,6))])
```

```
    >>> grade("This is my house.", "This is mi hhouse")
    (True, "typo", [((8,10), (8,10)), ((11, 16),(11,17))])

    >>> grade("This is my house.", "This my house!")
    (False, "missing", [((5,7), (5,5))])

    >>> grade("This is my house.", "This is your house!")
    (False, "wrong_word", [((8,10), (8,12))])

"""

## Punctuation

Punctuation (,?,!, etc.) should be ignored when grading.

## Typo detection

The student's answer can be at most edit distance 1 off from the correct
answer for each word as long as the student's word is not a valid word in English.
Swaps ("ae"/"ea") should count as edit distance 1.

For example:

    # edit distance 1 for each word
    >>> grade("The man eats the cheese.", "Thhe maan eatss thhe chheese")
    (True, "typo", [...])

    # "thhhe" is edit distance 2 from "the"
    >>> grade("The man eats the cheese.", "Thhe maan eatss thhhe chheese")
    (False, "wrong_word", [...])

    # "housed" is edit distance 1 but is a valid English word
    >>> grade("house", "housed")
    (False, "wrong_word", [...])

## Missing word detection

If exactly one word is missing in the student's answer, grade() needs to highlight it.

    # "is" is missing
    >>> grade("This is my house.", "This my house!")
    (False, "missing", [((5,7), (5,5))])
```

```
# 2 words missing
>>> grade("This is my house.", "This house!")
(False, None, [])
```

Wrong word detection

If exactly one word is wrong in the student's answer that word should be highlighted.

```
>>> grade("This is my house.", "This is your house!")
(False, "wrong_word", [((8,10), (8,12))])

# 2 words wrong
>>> grade("That is my house.", "This is your house!")
(False, None, [])
```

Unicode Support

Make sure your function understands unicode characters.

```
>>> grade(u"über is not an English word", u"über is an English word")
(False, "missing", [((8,11), (8,8))])
```

Hand-In

The source code (and installation instructions if necessary).

Evaluation

Your code is going to be evaluated by the following criteria:

1. Correctness:
   Does grade() work correctly for common inputs?
2. Completeness:
   Does grade() work for corner cases?
3. Cleanliness:
   Is the code well organized and is it easily readable?
4. Performance:
   Is the code fast?