22C:16 (CS:1210) Quiz 6

You have 20 minutes to complete this quiz.

1. We want to write a function `notNeighbors` that takes two words `word1` and `word2` as parameters and returns `True` if `word2` cannot be obtained from `word1` by substituting exactly one letter by another and returns `False` otherwise. Here is partial code to do this and you need to complete this code. Our plan is to count the number of corresponding letters in the two words that are not identical. If this count is not equal to 1 then the two words are not "neighbors" and the function should return `True`. The code given below is missing some lines and your task is to supply the missing code and complete the function. You don't have to check if the words contain non-letters – assume that they don't.

```
def notNeighbors(word1, word2):
    if len(word1) != len(word2):
        return True

    count = 0
    # Three lines of missing code for scanning the two words
    # and comparing corresponding letters goes here.









    if count != 1:
        return True
    else:
        return False
```

2. Define a function called `expand` that takes two lists – a list of words called wordList and a list of corresponding frequencies called `freqList` and returns an “expanded” list in which each word appears as many times as its frequency. For example, if `wordList` is `["hi", "hello", "prediction"]` and `freqList` is `[3, 2, 1]`, then the function should return `["hi", "hi", "hi", "hello", "hello", "prediction"]`. The “expanded” list should contain words in order, i.e., since `"hi"` appears first in `wordList`, all copies of `"hi"` should appear first in the “expanded” list, etc. You can assume that `freqList` contains positive integers only.