1. [5 points] What does each of the following expression evaluate to? Suppose that L is the list 
["These", "are", "a", ["few", "words"], "that", "we", "will", "use"].(a) L[3:4][0][1][2]
L[3:4] = [["few", "words"]]
L[3:4][0] = ["few", "words"]
L[3:4][0][1] = "words"
L[3:4][0][1][2] = 'r'

(b) "few" in L
False. The string "few" is part of a list which is an element of the list L, the literal string is not an element of L.

c) [L[1]] + L[3]
[L[1]] = ["are"]
L[3] = ["few", "words"]
[L[1]] + L[3] = ["are", "few", "words"]

d) L[4:]
["that", "we", "will", "use"]

e) L[0::2]
["These", "a", "that", "will"]

Turn over for Problem 2.
2. [5 points] Here is a partially completed function called \texttt{concatenate} that takes a list of strings as a parameter and returns a long string that is the concatenation of all the strings in the list, taken in order. For example, if the given list is ["These", "are", "hello"] then the function would return "Thesearehello". There is one line missing in this function. Your task is to supply this line.

```python
def concatenate(L):
    bigString = ""
    for i in range(len(L)):
        # Fill in the blank line below
        bigString = bigString + L[i]
                           -------------------------------
    return bigString
```