CSC 120 **ICA-42**

Work with your neighbor. (This will be graded for participation only.)

1. Write a function `next_words(w1, words_list)` that takes a word, `w1`, and a list of valid words, `words_list`, and returns a list of words that differ from `w1` by one letter. Add only valid words that appears in `words_list`.

2. Write a function `dist(w1, w2)` that returns the number of positions where words `w1` and `w2` differ.
   Requirements:
   - Use an assert to verify the lengths of `w1` and `w2` are the same.
   - Use a list comprehension in your function.

Final exam review problems

3. **Recursion**. Write a recursive function `count_occurrences(alist, value)` that returns the number of the occurrences of `value` in `alist`.

4. **LinkedLists**. Write a method `insert_after_pos(self, node, pos)` for the `LinkedList` class that adds the node `new` after position `pos`. Node positions begin at 0, i.e., the first node in the list has position 0. You can assume that the list will **NOT** be empty and that `pos >= 0`. If `pos` is greater than the length of the list, add `new` to the end of the list.