Project 3 FAQ

1. How do I find all the permutations of a string?

You can use this pseudocode which prints out all permutations of a string to get started:

```
void printPermutations(string prefix, string rest) {
     if (rest is empty) {
          Display the prefix string.
     }
     else {
          For each character in rest {
               Add the character to the end of prefix.
               Remove character from rest.
               Use recursion to generate permutations with the
updated values for prefix and rest.
          }
     }
}
```

2. I thought you said that we couldn't use any iteration. Isn't the pseudocode using a loop?

Yes, printPermutations uses iteration (a for loop), which this assignment does not allow. Suppose you have a for loop of the form:

```
for (i = initval; i < maxval; i++) {
     statement;
}
```

Consider the following recursive function:

```
void Loop(int i, int max) {
     if (i >= max)
          return;
     statement;
     Loop(i + 1, max);
}
```

You can replace the for loop above with the following invocation of Loop:

```
  Loop(initval, maxval);
```

3. **My program seems to work for small dictionaries, but crashes when it attempts to read in your large dictionary. What's happening?**

You probably didn't increase your stack size enough. Make sure you followed the instructions for increasing it on whatever platform you're using and if that still doesn't help, try a larger stack size.

4. **Do I have to use the above algorithm to do this assignment?**

No, you can use a different approach, if you like. However, make sure you understand how to break problems down recursively, that's what you'll be tested on during the exams.

5. **Why does my program work for small words but not for larger words?**

One possibility is that it actually does work for larger words, but just takes a really long time. We're not testing your programs for efficiency yet so it should be OK. If you're consistently having to wait long times for 5 or less letter words, you're probably doing something inefficiently (besides using recursion for this project).