Help on Recursion

Hello everyone! Since I will not be able to help you on the recursion programming assignment (Program #8) in class, I am writing this instead. I would like to suggest some recursive solutions to a couple of the problems. Hopefully, you will be able to complete the coding yourself.

1 Encryption

This problem is very similar to the `printDigits` method we did in class in that it uses the `/` and `%` operators. Let \( n \) be a nonnegative integer. Then, consider the following recursive solution:

\[
\text{encrypt}(n) = \begin{cases} 
\text{the letter corresponding to } n, & \text{if } n/10 = 0 \\
\text{encrypt}(n/10) + \text{letter corresponding to } n \mod 10, & \text{if } n/10 \neq 0
\end{cases}
\]

You should do a couple of simulations by hand to make sure you understand the solution. Basically, the front end of the number is being encrypted recursively, and then the rightmost digit is being converted to a letter and then concatenated with the encrypted front end of the number. (Notice that the `+` means concatenation.)

2 Letter Pattern

This one is really easy! Suppose \( \text{letter} \) is a character representing a capital English letter.

\[
\text{letterPattern}(\text{letter}) = \begin{cases} 
\text{output } \text{letter}, & \text{if } \text{letter} = \text{‘A’} \\
\{ \text{output } \text{letterPattern}(\text{previous letter}), \\
\text{output } \text{letter}, \\
\text{output } \text{letterPattern}(\text{previous letter}) \}, & \text{if } \text{letter} \neq \text{‘A’}
\end{cases}
\]

See the handout to find out how to obtain the previous letter using `char to int` conversion and then back!

3 Longest Common Subsequence

This one looks a lot harder than it really is! In fact, the recursive solution is given in the description. In what follows, let \( u \) denote the string obtained from \( \text{str1} \) by deleting the first character, and let \( v \) denote the string obtained from
\( str2 \) by deleting the first character.

\[
\text{longComSubSeq}(str1, str2) = \begin{cases} 
0, & \text{if either } str1 \text{ or } str2 \text{ is empty or null} \\
1 + \text{longComSubSeq}(u,v), & \text{if } str1 \text{ and } str2 \text{ have the same first character} \\
\max\{\text{longComSubSeq}(str1,v), \text{longComSubSeq}(u,str2)\} & \text{if } str1 \text{ and } str2 \text{ have different first characters}
\end{cases}
\]

4 Garbanzo Bean Game and Fractal Image

You’re on your own!