1: Dancing Words: Due September 4, 2015

CSCI 2212

1 Goals

1. To use an array of characters.
2. To read a line of text from the keyboard.
3. To use functions from the ctype library to do text manipulation.
4. To implement some simple but nontrivial logic.

Dancing Words. Rhythm is an important part of dancing. A dancer sets a rhythm (strong and weak beats) and sticks to it. We will say that word dances if it has strong and weak letters. For example,

A 2-step: *-*-*-
2 happiness
HaPpInEsS

A waltz: *_*_*_
3 waltzrhythm
3 WalTzrHytHm)

A march: *__*__*__
marchtothedrummer
MarcHtotHedrUmmeR

2 Instructions

1. Call banner() and bye() from main. Do not try to do the whole program in main!
2. Write a function to read and process one line of text. Write any other functions you need.
3. Input will come from the keyboard. Quit when the user enters a line that starts with `#`.
4. Each input line will contain a number between 2 and 4 followed by a space and a word or a line of nonsense (< 80 chars). Create an array of 80 characters to store it. The number defines the rhythm for that line.
5. Read each line using a single call on `scanf()` to read both the number and the text.
6. If the digit is not a legal value (2, 3, or 4) print an error comment and ignore the text.
7. Write a loop to process your string from the first char to the null character on the end.
8. Use toupper() and tolower() to make the letters follow the rhythm pattern.
9. If the line contains any character that is not a letter, keep it unchanged.
10. Print the original input sentence and print your line of dancing words on the next line. Leave a blank line before the next input/output pair.
11. Be sure your program handles a string with zero, one or two chars gracefully. Don't crash or output nonsense.

Testing and Output. Test your program with words of various lengths and various rhythms. Hand in the C code and the output.