Tutorial 1: Regular Expressions and Text Normalization

Q1. Write regular expressions for the following languages. By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

1. The set of all alphabetic strings;
2. The set of all lower case alphabetic strings ending with a letter b;
3. The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
4. All strings that start at the beginning of the line with an integer and that end at the end of the line with a word;
5. All strings that have both the word "grotto" and the word "raven" in them (but not, e.g., words like "grottos" that merely contain the word "grotto");

HINT: Not all notions are covered in lectures, and it is fine that your RE cannot fully satisfy the specified requirements.

Q2. Try all your answers on http://regexr.com/ You may need to change the textbox to test two cases: the textbox contains one or more matched strings, and the textbox does not contain any matched string. What are the errors (e.g., false positive and false negative) have you observed?

Q3. Select all strings that can be matched by regular expression /E*F+[^Gg]/

A. EFG B. EF C. FFF D. EFFa

Q4. Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.

Q5. Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of two sentences "computed the edit distance" to "the edit distance is computed". Show your work and show the alignment between the two strings. You may use edit distance defined at **word level** instead of character level.