1. (3 points each) Assume that you're using an open hash table and decide to rehash to a table that's twice as large whenever the total number of elements in the table's linked lists increases to twice the size of the table, in an effort to keep access time reasonable. To balance this technique, you decide to conserve memory by rehashing to a table that's half as large whenever the total number of elements in the table's linked lists decreases to half the size of the table.

If you assume that insertions into the hash table occur as frequently as deletions from the table, then it is important that you try to avoid either type of table rehashing as much as possible. Under these circumstances, explain how the methodology described above inappropriately tends to use one of the rehashing functions excessively.

2. Devise a simple, easy-to-calculate, well-distributed hash function for mapping two-letter words to the integers from 0 through 10. Restrict the variables used in your function to the ordinal values of the alphabetic characters in the words (i.e., A=1, B=2, C=3, etc.).

a. (3 points) Specify such a hash function that guarantees that no hash value is mapped to by more than two of the following two-letter words:

   AN AS AT BE DO GO HE IF IS IT ME NO ON SO TO WE

b. (3 points) Map all of the "legitimate" two-letter words to your hash function. Explain whether or not your hash function is a good one for hashing these thirty-two values. (Note: there are only 32 legitimate two-letter words: the sixteen above, as well as AD, AM, AX, BY, HI, ID, IN, MA, MY, OF, OR, OX, PA, PI, UP, and US.)

c. (3 points) Map the fifty two-letter state abbreviations to the integers from 0 through 10 via your hash function. Explain whether or not your hash function is a good one for hashing these fifty values. (Note: the two-letter state abbreviations are: AL, AK, AZ, AR, CA, CO, CT, DE, FL, GA, HI, ID, IL, IN, IA, KS, KY, LA, ME, MD, MA, MI, MN, MS, MO, MT, NE, NV, NH, NJ, NM, NY, NC, ND, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VT, VA, WA, WV, WI, and WY.)

3. (3 points) When deleting values from a hash table, it is important not to hinder later searches or to make any table slots unusable. One approach is to mark each slot from which an element was deleted as dead, indicating that it once contained a value, but does so no longer. When using this approach, explain what should be done when a dead slot is encountered in a search for a hashed value, and explain what should be done when a dead slot is encountered during the course of a hash table insertion.

You must provide your own solutions to these problems in a clearly presented Word document. Obtaining solutions from any outside source is considered academic misconduct. The only person with whom you may discuss these problems is the instructor.