Homework #11

Question 1. Please make a Turing machine with the alphabet $\{0, 1\}$ which computes the bitwise OR of two strings given as stacked inputs.

Question 2. Please make a Turing machine which converts separated strings to a stacked string on the alphabet $\{0, 1\}$ with separator symbol #. For example, your TM might start with something like 0010#0101 on the tape, and it would end with $\substack{0\\0}\substack{0\\1}\substack{1\\0}\substack{0\\1}$. You may assume that the two separated strings have the same length and are not empty.

Question 3. Please make a Turing machine which converts a stacked string to separated strings on the alphabet $\{0, 1\}$ with separator symbol #.

Question 4. Please explain (in ordinary words- doesn't need to be a real proof) how you could use the two TMs above, to make a TM which adds two binary numbers which are given as separated strings.

Question 5. Make a Turing Machine M which accepts all strings of the form xa, halts and rejects all strings of the form xab, and loops forever on all strings of the form xbb. (I don't care what happens if the input has less than 2 letters.)

Question 6. Explain why the set of recursive languages is closed under intersections. (You don't need a super-technical proof.) Hint: don't discuss specifc TMs: use the fact that recursive languages L are those for which there is an algorithm which can verify whether or not some string x is in L.

Question 7. Explain why the set of recursive languages is closed under complements. Explain specifically why your argument does not work for RE languages.

Question 8. Explain why: If L is RE but not recursive, then the complement $\bar{L}$ is not RE. *Hint*: Argue by contradiction. So you will assume that L is RE but not recursive, and also assume that $\bar{L}$ is RE, and explain why these two things are contradictory.