CAS CS 538. Problems on Multi-Party Computation and Secret Sharing

This is not a homework assignment; it's a set of problems to test your own understanding of the material

Problem 1. We defined security of two-party computation in terms of simulators. Describe the simulator for the honest-but-curious Oblivious Transfer (OT) functionality described in class.

Problem 2. NAND is complete for boolean circuits. Show how to implement an honest-but-curious NAND gate (where inputs come in shared and output is produced shared) using OT.

Problem 3. Write out explicitly the protocol that results from the GMW conversion of the honest-but-curious OT to the OT secure against malicious parties (note that both sides could be malicious here).

Problem 4. You have shares t_1, t_2, t_3 of a secret s modulo a prime p, shared using (3, 7) secret sharing. Write out an explicit formula for s (hint: lookup Lagrange Interpolation).

Problem 5. Suppose in (k, n) secret sharing, all n people came back to reconstruct the secret. Some gave their shares honestly, others dishonestly. You don't know which is which. Prove that if there are at least $(n+k)/2$ honest ones, then the secret is uniquely defined, anyway. (Hint: suppose there are two polynomials of degree $k-1$ that agree with $(n+k)/2$ of the shares. Use the fact that a polynomial of degree $k-1$ cannot have more that $k-1$ roots (why?) to get a contradiction.) FYI: there are algorithms to efficiently find this unique secret; see here for one such algorithm `http://www.cs.bu.edu/~reyzin/code/WelchBerlekamp.cpp`, although more efficient ones exist.