Problems 1-6 are 1 point each

1. Text Problem 7.9 p.256

2. Text Problem 7.12 p. 256

3. Text, Problem 8.6 p.300

4. Text, Problem 8.8 p.300. If you work this out correctly, the answer should strike you as rather odd and possibly even wrong. What is strange about this result?

5. Text, Problem 8.17 p.301

6. Text, Problem 8.19 p.302

7. (4 pts) Assuming that a, b and c are arrays of 1024 integers and j is an integer, translate the C fragment below into x86 assembler using each the following different types of addressing:
   a. based (e.g., [ebx])
   b. based-indexed (e.g. [ebx+esi])
   c. based with displacement (e.g., [ebx +aj])
   d. scaled-indexed-based (e.g., [ebx + ecx * 4 + a])

   for (j = 0; j < 1024; j++) {
     a[j] = b[j] + c[j];
   }

You can use either conditional jumps or a LOOP instruction for loop. You can write either 16 or 32-bit instructions.