ECE 468 & 573
Problem Set 3: Common sub-expression elimination and local register allocation

For the following problems, consider the following piece of three-address code:

1. A = 7;
2. B = A + 2;
3. C = A + B;
4. D = A + B;
5. A = C + B;
6. B = C + B;
7. E = C + D;
8. F = C + D;
9. G = A + B;
10. H = E + F;
11. I = H + G;
12. WRITE(I);

1. Show the result of performing Common Subexpression Elimination (CSE) on the above code.
2. Suppose A and C were aliased. How would that change the results of CSE?
3. For each instruction, show which variables are live immediately after the instruction.
4. How many registers would be needed to perform register allocation with no spilling?
5. Top down register allocation is inefficient for the above code, as there are some variables that could safely be assigned to the same register. What are they?
6. Perform bottom-up register allocation on the code for a machine with four registers. Show what code would be generated for each 3AC instruction. When choosing registers to allocate, always allocate the lowest-numbered register available. When choosing registers to spill, choose the non-dirty register that will be used farthest in the future. In case all registers are dirty, choose the register that will be used farthest in the future. In case of a tie, choose the lowest-numbered register.

Repeat steps 3 and 6 for the following code, but with three registers (assume registers can hold either temporaries or variables):

1. T1 = A + B;
2. T2 = C + D;
3. T3 = T1 - T2;
4. T4 = C + T3;
5. \( T_5 = D + T_3; \)
6. \( T_6 = T_1 + T_5; \)
7. \( T_7 = D + T_6; \)
8. \( T_8 = B + T_7; \)
9. \( T_9 = A + T_4; \)
10. \( A = T_1 + T_8; \)
11. \( T_{10} = T_9 + A; \)
12. WRITE(T_{10});