In LC3 assembly, write fragments of code to do the following.... If you need a "junk" register, use register 4.

1) \( R_0 = 2 \cdot R_3 - R_2; \)

2) if (\( R_1 \) is odd) 
   \( R_2 = R_2 + 8; \)
   else 
   \( R_5 = R_5 + R_2; \)

3) \( r_0 = 10; \)
   \( r_1 = 5; \)
   while (\( r_0 > 0 \)) 
   { 
     \( r_1 = r_1 + 3; \)
     \( r_0 = r_0 - 1; \)
   }

4) \( R_0 \) is a pointer to a STRINGZ of ASCII characters (terminated by a zero). Write LC3 code that adds up the ASCII values, leaving the result in \( R_1 \).

5) Write an LC3 subroutine called \( \text{SUB1} \) that adds 4 to \( R_5 \), and then returns.

6) Write an LC3 subroutine called \( \text{SUB2} \) that doubles the value in \( R_5 \), and then calls \( \text{SUB1} \) before returning. WRITE THIS CODE IN THE SIMULATOR, AND MAKE SURE IT WORKS!

7) At memory location \( x_{3000} \), there are the following four hex values:
   \( x_{5020} \)
   \( x_{1025} \)
   \( x_{23FD} \)
   \( x_{5401} \)
   Convert each of these to LC3 assembly code.
   If you execute these four instructions, what values are in \( R_0 \), \( R_1 \), and \( R_2 \) (show your answers for the registers in hex).
   TRY THIS OUT ON THE SIMULATOR. SERIOUSLY, DO IT.

8) For the following code, convert the first two lines to machine code, and then into hex. What values are in \( R_0 \), \( R_7 \) (that's \( R_7 \), not \( R_1 \)), when you get to \( \text{HALT} \). What values are in the memory locations \( X \), \( Y \), and \( Z \).

   \[ \text{LEA R0, FOO} \]
   \[ \text{JSR FOO} \]
   \[ \text{HALT} \]

   \( \text{FOO} \)
   \[ \text{ADD R1, R0, #6} \]
   \[ \text{LDI R0, X} \]
   \[ \text{STR R0, R1, #-2} \]
   \[ \text{RET} \]

   \( X \)
   \[ .\text{FILL Y} \]

   \( Y \)
   \[ .\text{FILL X27} \]

   \( Z \)
   \[ .\text{FILL X37} \]

   \[ .\text{END} \]
### Figure A.2
Format of the entire LC-3 instruction set.

**Note:** + indicates instructions that modify condition codes.