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

1) \( R0 = R3 \) - \( R2 \);

2) if (R1 is even)
   \[ R2 = R2 + 8; \]
else
   \[ R5 = R1 + R2; \]

3) \( R0 \) 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 R1.

4) Write an LC3 subroutine called SUB1 that adds 5 to R0, and then returns.

5) Write an LC3 subroutine called SUB2 that doubles the value in R0, and then calls SUB1 before returning.

6) Using the PUTS trap function, write a complete LC3 program that will print out "Hello World." Include the .ORIG, .END, and so on.

7) The ASCII value for capital \( \text{A} \) is \( x41 \). Translate the following:
\[ x45 \ x4E \ x49 \ x47 \ x4D \ x41 \]

8) The OUT trap vector sends a single letter (in R0) to the screen, using memory mapped I/O. Write some LC3 code that will perform this trap function. In case you forgot, the display status register gives you a negative number when the display is ready. To get you started, here's a little bit of useful information:
\[
\text{OS\_DSR} \quad .\text{FILL} \ xFE04 \nn\text{OS\_DDR} \quad .\text{FILL} \ xFE06
\]

9) At memory location \( x3000 \), there are the following four hex values:
\[ x5020 \nn\ x1025 \nn\ x23FD \nn\ x5401 \]
Convert each of these to LC3 assembly code.
If you execute these four instructions, what values are in R0, R1, and R2 (show your answers for the registers in hex).
10) For the following code, what values are in R0, R1, when you get to halt. What values are in the memory locations X, Y, and Z.

```
.ORIG x3000
   LEA R0, X
   LD  R1, Y
   STI R1, X
   ADD R1, R1, #3
   STR R1, R0, #1
   HALT
X  .FILL Z
Y  .FILL x51
Z  .FILL x75
.END
```

| R0: | R1: | X: | Y: | Z: |

Show the symbol table for the code above

Using the LC3 reference sheet, change each line of the program to hex

11) For the following code, what values are in R0, R7 (that's R7, not R1), when you get to halt. What values are in the memory locations X, Y, and Z.

```
.ORIG x3000
   LEA R0, FOO
   JSR FOO
   HALT
FOO    ADD R1, R0, #6
       LDI R0, X
       STR R0, R1, #-2
       RET
X      .FILL Y
Y      .FILL x27
Z      .FILL x37
.END
```

| R0: | R7: | X: | Y: | Z: |

Show the symbol table for the code above

Using the LC3 reference sheet, change each line of the program to hex