Problem 1:  (8 points)

(a) If initially R1 = -8 and R2 = 8, what is the value of the PC after the following instructions are executed?

\[
\begin{array}{ll}
x3008: & 0101 011 001 000 010 \\
x3009: & 0000 010 000 000 100 \\
\end{array}
\]

\text{AND r3 r1 r2}

\[
\begin{array}{ll}
x3008: & 0101 011 001 000 010 \\
x3009: & 0000 010 000 000 100 \\
\end{array}
\]

\text{brz pc + 4}

\[
PC = x300A
\]

(b) What is the final value of R2 after this program executes? (R2 = 1, R4 = 6 initially)

\[
x4001: 0001 010 010 000 010 \\
x4002: 0001 100 100 111 111 \\
x4003: 0000 011 111 111 101 \\
\]

\text{Add r2,r2,r2}

\text{Add r4,r4,}\#-1

\text{BRzp here}

128

Problem 2:  (8 points)

If the number of registers in LC3 is doubled, what would be the effect on:

(a) The ADD immediate instruction

\text{Imm is 3 bit: range -4 to 3}

(b) The range of addresses a JUMP instruction can have

\text{No Change}

Problem 3:  (8 points)

How will you implement the following operation in LC3?

R4 = R1 NOR R2

Write the machine code (binary 16 bit instructions) in the space below.

\[
\begin{array}{ll}
\text{NOT R1, R1} & 1001 001 001 111 111 \\
\text{NOT R2,R2} & 1001 001 001 111 111 \\
\text{AND R4,R1,R2} & 0101 100 001 000 010 \\
\end{array}
\]
Problem 4: (10 points)

Shown here are the contents of memory before and after the LC3 instruction at location x3010 is executed. Your job: Identify the instruction stored in x3010. Note: There is enough information below to uniquely specify the instruction at x3010.

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>x3208</td>
</tr>
<tr>
<td>R1</td>
<td>x2d7c</td>
</tr>
<tr>
<td>R2</td>
<td>x2053</td>
</tr>
<tr>
<td>R3</td>
<td>x33ff</td>
</tr>
<tr>
<td>R4</td>
<td>x3f1f</td>
</tr>
<tr>
<td>R5</td>
<td>x4a2</td>
</tr>
<tr>
<td>R6</td>
<td>x7a00</td>
</tr>
<tr>
<td>R7</td>
<td>x7a2b</td>
</tr>
<tr>
<td>x3400</td>
<td>x3001</td>
</tr>
<tr>
<td>x3401</td>
<td>x7a00</td>
</tr>
<tr>
<td>x3402</td>
<td>x7a2b</td>
</tr>
<tr>
<td>x3403</td>
<td>xa700</td>
</tr>
<tr>
<td>x3404</td>
<td>xf011</td>
</tr>
<tr>
<td>x3405</td>
<td>x2003</td>
</tr>
<tr>
<td>x3406</td>
<td>x31ba</td>
</tr>
<tr>
<td>x3407</td>
<td>xc100</td>
</tr>
<tr>
<td>x3408</td>
<td>xefef</td>
</tr>
</tbody>
</table>

x3010: STR R2, R4, #7 (0111010 100 000111 )

Problem 5: (6 points)

Explain in brief, the difference between:

(a) Data errors and logic errors

Data errors: Input data is different than expected
Logic errors: Program is legal but logically wrong

(b) Trap and jump instructions

Trap: Target address is obtained via a table, RET causes old PC to be restored
Jump: Target address specified by user through a register, PC is not saved

(c) Breakpoints and watchpoints

Breakpoints: Program stops executing where we set the breakpoint
Watchpoints: Execution stops when the variable being watched changes
Problem 6:  (8 points)
Let R1, R2 and R3 hold some values. Draw a flowchart to store the greatest of these values in R4. Your flowchart must consist of blocks which correspond to groups of LC3 instructions.

Problem 7:  (8 points)
We are about to execute the following program:

<table>
<thead>
<tr>
<th>Address</th>
<th>ISA Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>x3000</td>
<td>1110 0000 0000 1110 ; LEA R0, x00B</td>
</tr>
<tr>
<td>x3001</td>
<td>0010 0010 0000 1110 ; LD R1, x00D</td>
</tr>
<tr>
<td>x3002</td>
<td>0110 0100 1100 1110 ; LDR R2, R3, x0B</td>
</tr>
<tr>
<td>x3003</td>
<td>1111 0000 0010 0101 ; HALT</td>
</tr>
</tbody>
</table>

The state of the machine before the program starts is given below:

Memory location x300B contains x311B
Memory location x300C contains xC465
Memory location x300E contains xFE12
Memory location x300F contains x1243
Register R0 and R1 contain 0000
Registers R2 and R3 contain x3000
What will be the final contents of registers R0-R3 when we reach the HALT instruction? Write your answers in hexadecimal format.

\[
R0 = x300C \\
R1 = x1243 \\
R2 = x311B \\
R3 = x3000
\]

**Problem 8: (4 pts)**

Consider the following snippets of LC3 code which achieve the same function:

(i) \( \text{LDI R5, LABEL}_1 \) \text{ where } \text{LABEL}_1 = x4010

(ii) \( \text{LD R5, LABEL}_2 \) \text{ where } \text{LABEL} = x400F \\
\quad \text{LDR R5, R5, #1}

What is the advantage of using (i) over (ii)?

*More compact*

What is the advantage of using (ii) over (i)?

*More flexible because of the offset*