Problem 1: (8 points)
(a) If initially R1 = 3 and R2 = -3, what is the value of the PC after the following instructions are executed?

\[
\begin{align*}
x_{3018}: & \quad 0101 \ 011 \ 001 \ 000 \ 010 \quad \text{AND r3 r1 r2} \\
x_{3019}: & \quad 0000 \ 010 \ 000 \ 000 \ 100 \quad \text{brz pc + 4}
\end{align*}
\]

\[
PC = x_{301A}
\]

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

\[
\begin{align*}
x_{4001}: & \quad 0001 \ 010 \ 010 \ 000 \ 010 \quad \text{Add r2,r2,r2} \\
x_{4002}: & \quad 0001 \ 100 \ 100 \ 111 \ 111 \quad \text{Add r4,r4,} \#-1 \\
x_{4003}: & \quad 0000 \ 001 \ 111 \ 111 \ 101 \quad \text{BRp here}
\end{align*}
\]

64

Problem 2: (8 points)
If the number of registers in LC3 is doubled, what would be the effect on:

(a) The offset field of the LEA instruction

*Offset becomes 8 bit: -128 to 127*

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

*No Change*

Problem 3: (8 points)
How will you implement the following operation in LC3?

\[
R2 = R3 \ \text{NOR} \ R4
\]

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

\[
\begin{align*}
\text{NOT R3, R3} & \quad 1001 \ 011 \ 011 \ 111 \ 111 \\
\text{NOT R4, R4} & \quad 1001 \ 100 \ 100 \ 111 \ 111 \\
\text{AND R2, R3, R4} & \quad 0101 \ 010 \ 011 \ 000 \ 100
\end{align*}
\]
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></th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>x3208</td>
<td>x3208</td>
</tr>
<tr>
<td>R1</td>
<td>x2d7c</td>
<td>x2d7c</td>
</tr>
<tr>
<td>R2</td>
<td>xe373</td>
<td>xe373</td>
</tr>
<tr>
<td>R3</td>
<td>x2053</td>
<td>x2053</td>
</tr>
<tr>
<td>R4</td>
<td>x33ff</td>
<td>x33ff</td>
</tr>
<tr>
<td>R5</td>
<td>x3f1f</td>
<td>x3f1f</td>
</tr>
<tr>
<td>R6</td>
<td>xf4a2</td>
<td>xf4a2</td>
</tr>
<tr>
<td>R7</td>
<td>x5220</td>
<td>x5220</td>
</tr>
</tbody>
</table>

x3010: STR R2, R4, #8 (01110101 00100000)

Problem 5: (6 points)

Explain in brief, the difference between:

(a) 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

(b) Breakpoints and watchpoints

**Breakpoints:** Program stops executing where we set the breakpoint

**Watchpoints:** Execution stops when the variable being watched changes

(c) Data errors and logic errors

**Data errors:** Input data is different than expected

**Logic errors:** Program is legal but logically wrong
Problem 6:  (8 points)

Let $R_1$, $R_2$ and $R_3$ hold some values. Draw a flowchart to store the greatest of these values in $R_5$. 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 R1, x00B</td>
</tr>
<tr>
<td>x3001</td>
<td>0010 0010 0000 1110 ; LD R0, 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 x0000  
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.  

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

**Problem 8: (4 pts)**

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

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

(ii)  
\[ \text{LD R5, LABEL\_2} \quad \text{where LABEL = x400E} \]
\[ \text{LDR R5, R5, #2} \]

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*