Problem 1:  (8 points)
(a) If initially $R_1 = -4$ and $R_2 = 4$, what is the value of the PC after the following instructions are executed?

\[
\begin{align*}
x_{3028} & : & 0101 & 011 & 001 & 000 & 010 & \text{AND } r3 \ r1 \ r2 \\
x_{3029} & : & 0000 & 010 & 000 & 000 & 100 & \text{brz } pc + 4
\end{align*}
\]

$PC = x_{302A}$

(b) What is the final value of $R_2$ after this program executes? ($R_2 = 2$, $R_4 = 5$ initially)

\[
\begin{align*}
x_{4001} & : & 0001 & 010 & 010 & 000 & 010 & \text{Add } r2, r2, r2 \\
x_{4002} & : & 0001 & 100 & 100 & 111 & 111 & \text{Add } r4, r4, #\ -1 \\
x_{4003} & : & 0000 & 011 & 111 & 111 & 101 & \text{BRzp here}
\end{align*}
\]

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

*Imm is 3 bit: range -4 to 3*

(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?
$R_4 = R_5 \text{ NOR } R_6$

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

\[
\begin{align*}
\text{NOT } R_5, R_5 & : & 1001 & 101 & 101 & 111 & 111 \\
\text{NOT } R_6, R_6 & : & 1001 & 110 & 110 & 111 & 111 \\
\text{AND } R_4, R_5, R_6 & : & 0101 & 100 & 101 & 000 & 110
\end{align*}
\]
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>
<tr>
<td></td>
<td>x3400</td>
<td>x3001</td>
</tr>
<tr>
<td></td>
<td>x3401</td>
<td>x7a00</td>
</tr>
<tr>
<td></td>
<td>x3402</td>
<td>x7a2b</td>
</tr>
<tr>
<td></td>
<td>x3403</td>
<td>xa700</td>
</tr>
<tr>
<td></td>
<td>x3404</td>
<td>xf011</td>
</tr>
<tr>
<td></td>
<td>x3405</td>
<td>x2003</td>
</tr>
<tr>
<td></td>
<td>x3406</td>
<td>x31ba</td>
</tr>
<tr>
<td></td>
<td>x3407</td>
<td>xc100</td>
</tr>
<tr>
<td></td>
<td>x3408</td>
<td>xefef</td>
</tr>
</tbody>
</table>

**x3010: STR R2, R4, #9 (0111010 100 001001 )**

Problem 5:  (6 points)

Explain in brief, the difference between:

(a) Breakpoints and watchpoints

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

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

(b) Data errors and logic errors

*Data errors: Input data is different than expected*

*Logic errors: Program is legal but logically wrong*

(c) 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*
Problem 6: (8 points)

Let R1, R2 and R3 hold some values. Draw a flowchart to store the smallest 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 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.

\[
\begin{align*}
R0 &= \text{x}311B \\
R1 &= \text{x}1243 \\
R2 &= \text{x}300C \\
R3 &= \text{x}3000 \\
\end{align*}
\]

**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 = x400D}
\text{LDR R5, R5, \#3}

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*