C to LC3 Arrays: Examine the code segments below. What are the LC3 instructions generated by a C compiler for the C code below. The symbol table is shown below...

Symbol Table

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Type</th>
<th>Offset</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>grid</td>
<td>Int</td>
<td>-9</td>
<td>foo</td>
</tr>
<tr>
<td>x</td>
<td>Int</td>
<td>-10</td>
<td>foo</td>
</tr>
<tr>
<td>ptr</td>
<td>Int</td>
<td>-11</td>
<td>foo</td>
</tr>
</tbody>
</table>

Recall: R5 is frame pointer (dynamic link), R6 is top of stack, R4 is pointer to static area.

grid[6] = 5;
AND R0, R0, #0; set value of 5 in a register
ADD R0, R0, #5 ; R0 = 5
ADD R1, R5, #-9 ; get address of grid[0]
ADD R1, R1, #6 ; add 6 to that address
STR R0, R1, #0 ; store 5 into that address

x = grid[3] +1;
ADD R1, R5, #-9 ; get address of grid[0]
ADD R1, R1, #3 ; add 3 to the address
LDR R0, R1, #0 ; fetch value at grid[3]
ADD R0, R0, #1 ; add 1 to it
STR R0, R5, # -10; store into x

grid[x+2] = grid[x] +2;
LDR R0, R5, #-10 ; get value of x into register R0
LDR R1, R5, #-9 ; get address grid[0] into R1
ADD R1, R1, R0 ; and add x to it
LDR R2, R1, #0 ; load value at grid[x]
ADD R2, R2, #2 ; store this value into grid[x+2]
ADD R1, R1, #2 ; recall R1 contains address; grid[x] from earlier
STR R2, R1, #0 ; add 2 to value of x
; add this to grid[0] address
; store into this address

ptr = grid;
ADD R0, R5, #-9 ; address of grid
LDR R0, R5, #-11 ; set ptr equal to grid[0]