1. (5 points) Consider the following pair of eight bit strings: 0011 0101 and 0011 0100. If you interpret these as eight bit binary values, what is their sum?

If you interpret the given values as ASCII characters codes, what is the ASCII character that corresponds to their sum (hint: the ASCII code for ‘0’ is 30\text{_{16}}).

If you interpret the given values as BCD values using the excess-3 variant of BCD, what is the bit pattern for the sum of the given excess-3 values?
2. (5 points) If our basic processor begins execution with the first instruction shown below, what value will be in the accumulator after two instructions are executed? (Hint: sign-extension.) The list of instructions is given below.

<table>
<thead>
<tr>
<th>addr</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0c38</td>
<td>1ffe</td>
</tr>
<tr>
<td>0c39</td>
<td>ac3b</td>
</tr>
<tr>
<td>0c3a</td>
<td>4c3b</td>
</tr>
<tr>
<td>0c3b</td>
<td>9c3a</td>
</tr>
</tbody>
</table>

What value will be in the program counter after four instructions are executed?

0000  halt – halt execution
0001  negate – \( ACC := -ACC \)
1xxx  immediate load – if sign bit of xxx is 0 then \( ACC := 0xxx \) else \( ACC := fxxx \)
2xxx  direct load – \( ACC := M[0xxx] \)
3xxx  indirect load – \( ACC := M[M[0xxx]] \)
4xxx  direct store – \( M[0xxx] := ACC \)
5xxx  indirect store – \( M[M[0xxx]] := ACC \)
6xxx  branch – \( PC := 0xxx \)
7xxx  branch if zero – if \( ACC = 0 \) then \( PC := 0xxx \)
8xxx  branch if positive – if \( ACC > 0 \) then \( PC := 0xxx \)
9xxx  branch if negative – if \( ACC < 0 \) then \( PC := 0xxx \)
axxx  add – \( ACC := ACC + M[0xxx] \)