a. (2 pts) Give the 8-bit 2’s complement hex value for –25.

Convert +25 to hex. +25 = 0x19. Subtract +25 from zero to get –25.
0 – (+25) = 0x00 – 0x19 = 0xE7

b. (2 pts) The value 0xE1 is an 8-bit 2’complement number; give its decimal value.

The MSb of 0xE1 is a ‘1’, so this is a negative number. Subtract from zero to get its magnitude. 0x00 – 0xE1 = 0x1F, so the number is a –31. Do not forget the negative sign!!!

c. (3 pts) Write the following in PIC18 assembly.

unsigned int r,p,q;

do {
    // loop body – just write placeholders here
    //
} while ((p != 0) && (q >= r))

loop_top:
    ...instr loop body..
    ;test p!=0
    movf   p,w    ;p  LSB
    iorwf  p+1,w   ;p MSB
    bz   loop_exit
    ;test q>= r
    movwf  r,w
    subwf  q,w    ;LSB (q-r)
    movwf  r+1,w
    subwfb q+1,w  ;MSB (q-r)
    bc     loop_top

loop_exit:
    ...instr...

For q >= r test:

<table>
<thead>
<tr>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>no Borrow</td>
<td>no borrow</td>
</tr>
<tr>
<td>C = 1</td>
<td>C=0</td>
</tr>
</tbody>
</table>

d. (3 pts) Assuming 2’s complement numbers, give the TEST and the flag conditions needed for both the TRUE and FALSE conditions for the test “ p >= q “.

For p >= q test:

<table>
<thead>
<tr>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive result</td>
<td>negative result</td>
</tr>
<tr>
<td>N=0, V=0</td>
<td>N=1,V=0</td>
</tr>
<tr>
<td>or</td>
<td>or</td>
</tr>
<tr>
<td>N=1, V=1</td>
<td>N=0,V=1</td>
</tr>
</tbody>
</table>