You may NOT use a calculator. Assume the following memory/register contents at the beginning of each instruction. For problems a/b give the final value of the affected W or memory location (you do not have to give status flag values). For problems c/d you may use temporary locations if required; name them as tmp1, tmp2, tmp3, etc.

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x059</td>
<td>0xA8</td>
</tr>
<tr>
<td>0x05A</td>
<td>0x08</td>
</tr>
<tr>
<td>0x05B</td>
<td>0x8D</td>
</tr>
<tr>
<td>0x05C</td>
<td>0x29</td>
</tr>
</tbody>
</table>

W = 0xE3, STATUS = 0x00

**bit clear f**

7654 3210

[0x59] = 0xA8= 1010 1000 (clr bit 5)

new value [0x59]= 1000 1000 = 0x88

a. (2 pts) bcf 0x059, 5

b. (2 pts) xorwf 0x5C,w

W reg = (0x05C)= 0x29 = 0010 1001
xor (w) = 0xE3 = 1110 0011
new W = 0xCA = 1100 1010

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

```c
unsigned char i,k;
if ( i && !k)
   //if body, placeholder
}
```

```assembly
movf i, f ; test if i is nonzero
bz end_if ;skip body if i is zero
movf k, f ; test if k is zero
bnz end_if ;skip if k is nonzero
...if body...
...if_body....
end_if
...rest of code....
```

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

```c
unsigned char i;
loop
   do {
      //loop body placeholder
   }while (i > 0x30)
```

```assembly
loop_top
   ....loop_body...
   ....loop_body...
movf i,w
sublw 0x30 ; do 0x30 – i
bnc loop_top ;true if borrow (C=0), loop
...rest of code....
```