Q1a) With 3 bits, how do we represent \(-2\)? If it can't be done, select "N/A". (Select ONE per row)

<table>
<thead>
<tr>
<th></th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsigned</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Sign/Magnitude</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td></td>
<td>+0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>-0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>One's Complement</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td></td>
<td>+0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-0</td>
<td></td>
</tr>
<tr>
<td>Two's Complement</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Bias; use bias of -(2^{n-1}-1) from lecture</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Q1b) Convert 26\textsubscript{10} to the following bases (and remove any leading zeros)

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b11010</td>
<td>0x1A</td>
</tr>
</tbody>
</table>

\[1\times16+1\times8+0\times4+1\times2+0\times1=26\]

\[1\times16+10 (=A) \times1=26\]

Q1c) Add these Two’s Complement nibbles:

\[
\begin{array}{c}
1001 \\
+ 1011 \\
\hline
10100
\end{array}
\]

Does it overflow a nibble? (Select ONE)

- Yes \[1001 (\text{-7}) + 1011 (\text{-5}) = \text{-12}. Since 2s complement nibbles can only represent [-8,7] it certainly can't hold -12. You don't have to even do the addition to answer this correctly\]
- No

```c
int mystery (unsigned int N) {
    unsigned int counter = 0;
    while (N > 0) {
        counter += N & 1;
        N = N >> 1;
    }
    return counter == 1;
}
```

Q2a) What does the mystery return? (Select ONE)

- The number of 1s in the binary representation of N
- 1 if N is odd, otherwise 0
- 1 if N is a power of 2, otherwise 0 [it shifts N to the right, storing in counter all the 1s it sees. If it's exactly 1, meaning the only 1 is the MSB (most significant bit), then it's a power of 2]
- 1 if the binary representation of N is all 1s, otherwise 0
- 1 if the binary representation of N has any 1s, otherwise 0

Q2b) Given this setup to mystery:

```c
unsigned int myN = GetNFromUser();
int mysteryReturn = mystery(myN);
...could myN be changed by the call to mystery? (Select ONE)
- Yes
- It depends on the value of myN
- No [because C is call by value]```
// My project partner wrote this code to duplicate some elements of orig into copy
int orig[] = {1,2,3,4,5,6,7,8}; // ints are 4 bytes wide
int main() {
    int *backup, *copy, **copyH;
    backup = copy = (int *) malloc (sizeof(int) * 100);
    copyH = &copy;
    for (int i = 0; i < 2; i++) {
        *copy = orig[i];
        *copyH = *copyH + 4;
    }
}

Q3a) Right before the for loop, where in memory do the following point? (Select ONE per row)

<table>
<thead>
<tr>
<th>Code</th>
<th>Static</th>
<th>Stack</th>
<th>Heap</th>
</tr>
</thead>
<tbody>
<tr>
<td>orig</td>
<td>◯</td>
<td>◯</td>
<td>◯ ◯</td>
</tr>
<tr>
<td>backup</td>
<td>◯</td>
<td>◯</td>
<td>◯ ◯</td>
</tr>
<tr>
<td>copyH</td>
<td>◯</td>
<td>◯</td>
<td>◯ ◯</td>
</tr>
</tbody>
</table>

Q3b) Right after the for loop, what is the value of the following? If it'd be garbage, write “G”

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
</tbody>
</table>

backup = copy = (int *) malloc (sizeof(int) * 100);
the memory below shows all the words (not bytes) of the malloced space

Q4a) Which RISC-V snippet could be the compilation of the C code: x15 = 20 - x5? (Select ALL that apply)

Assume the C variables x5 and x15 map directly to the registers of the same name.

- ▢ sub x5, 20, x15 [no, sub has no immediates]
- ▢ sub x15, 20, x5 [no, sub has no immediates]
- □ addi x15, x0, 20 #x15=20
- ▢ sub x15, x15, x5 #x15=x15-20
- ▢ addi x15,x5,-20 #x15=x5-20
- ▢ sub x15, x0, x15#x15=x15-20

Q4b) Say we have an int array A[99] starting at address 0x00010000, and register x5 contains &A[0]. Assuming sizeof(int) == 4, what value is in register x10 after lw x10, 8(x5)? (Select ALL that apply)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bytes from A is 2 ints over, thus A[2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>