#1. In a typical CISC architecture

A) Caller       B) Callee

_____ accesses the formal parameters via an offset from the FP (frame pointer).
_____ allocates space for local variables.
_____ allocates space for the return value.
_____ stores the return value in the return value location.
_____ saves the PC (program counter) as the return address.
_____ pushes the actual arguments onto the stack.

#2. a) Convert 127.875\textsubscript{10} to binary fixed-point and single precision IEEE floating-point representation (expressed in hexadecimal).

binary fixed-point __________________________________ x 2\textsuperscript{0}

IEEE floating-point __________________________________ (hexadecimal)

b) Convert 0xC2568000 (single precision IEEE floating-point representation) to fixed-point decimal.

fixed-point decimal __________________________________ (decimal / no exponential notation)
#3. Given

```c
static void fubar( int a )
{
    static int b = 7;
    void (*c) (int) = fubar;
    ...
}
```

When this function is called, identify which area of the C Runtime Environment each of the following will be allocated and its scope or visibility.

<table>
<thead>
<tr>
<th>Area of Runtime Env.</th>
<th>Scope/Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>____________</td>
</tr>
<tr>
<td>b</td>
<td>____________</td>
</tr>
<tr>
<td>c</td>
<td>____________</td>
</tr>
<tr>
<td>fubar</td>
<td>____________</td>
</tr>
</tbody>
</table>

Where c is pointing ____________

If the function above is called 5 times, indicate how many times will b be initialized to 7? _________

#4. What gets printed with the function call mystery( 10 );?

```c
int mystery( int param ) {
    int local = 30;

    if ( local > param )
    {
        local = local - param;
        printf( "%d\n", local );  /* Output the value of local followed by a newline */
        param = mystery( param + 5 ) + local;
        printf( "%d\n", param );  /* Output the value of param followed by a newline */
    } else {
        printf( "Whoa\n" );
    }

    return local;
}
```

Put answer here