#1. In a typical CISC architecture

A) Caller  

_____ 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.

_____ accesses the formal parameters via an offset from the FP (frame pointer).

_____ allocates space for local variables.

_____ allocates space for the return value.

B) Called/Callee


#2. a) Convert $137.875_{10}$ to binary fixed-point and single precision IEEE floating-point representation (expressed in hexadecimal).

binary fixed-point _____________________________ $ \times 2^0$

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
void fubar( int a )
{
    static int b = 42;
    int *c = &a;
    ...
}
```

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

<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 7 times, indicate how many times will `b` be initialized to 42? _________

#4. What gets printed with the function call `mystery( 5 );`?

```c
int mystery( int param ) {
    int local = 25;

    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( "Stop\n" );
    }

    return local;
}
```

Put answer here