#1. In a typical CISC architecture

1) Caller  
   2) Called/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.

Which instruction saves the current value of the Program Counter (%pc) into %o7 as the saved return address?

#2. a) Convert \(-124.375\)\(_{10}\) 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 \(0x42768000\) (single precision IEEE floating-point representation) to fixed-point decimal.

fixed-point decimal ___________________________ (decimal / no exponential notation)

(over)
**#3. Given**

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

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 b is pointing___________

If the function above is called 7 times, indicate how many times will c be initialized to 42? ________

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

```c
int mystery( int param ) {
    int local = 45;

    if ( local > param )
    {
        local = local - param;
        printf( "%d\n", local ); /* Output the value of local followed by a newline */
        param = mystery( param + 10 ) + local;
        printf( "%d\n", param ); /* Output the value of param followed by a newline */
    } else {
        printf( "Halt\n" );
    }

    return local;
}
```

Put answer here