#1. In the SPARC architecture:

If the caller passes 3 actual arguments to a function, those values as formal parameters are accessed in the callee in register ______ through ______.

The __________ instruction allocates space for local variables. These local variables are accessed with a __________ offset relative to register ______.

The callee places the return value into register ______ before executing ret/restore.

The _____ instruction saves the PC (program counter) as the return address into register ______.

The caller retrieves the return value from register ______ after the function it called returns.

#2. a) Convert 136.875\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 0xC25E000 (single precision IEEE floating-point representation) to fixed-point decimal.

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

```c
int a;
static void fubar( int b )
{
    static int c = 7;
    void (*d) (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 (Global/File/Function)</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>d</td>
<td>__________________</td>
</tr>
<tr>
<td>fubar</td>
<td>__________________</td>
</tr>
</tbody>
</table>

Where d is pointing __________

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

#4. What gets printed with the function call mystery( 10 ); ? (Hint: Draw stack frames!)

```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( "Whoa\n" );
    }

    return local;
}
```

Put answer here