#1. Given

```c
static int a = 10;
void fubar( int b )
{
    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 (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>fubar</td>
<td>___ _________</td>
</tr>
</tbody>
</table>

Where c is pointing ___ _________

If the function above is called 5 times, indicate how many times will a be initialized to 10? ________

Using the Rt-Lt Rule, declare bar to be a function that takes a pointer to a double and returns a pointer to an array of 42 elements where each element is a pointer to a struct foo.

#2. a) Convert 144.125₁₀ to binary fixed-point and single precision IEEE floating-point representation (expressed in hexadecimal).

- binary fixed-point ______________________________ x 2⁰
- IEEE floating-point ______________________________ (hexadecimal)

b) Convert 0xC21E8000 (single precision IEEE floating-point representation) to fixed-point decimal.

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

(over)
#3. What is the output of the following program? (Hint: Draw stack frames!)

```c
int mystery( int param )
{
    int local = 35;

    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;
}
```

#4. What gets printed with the function call `mystery( 5 );`?
( Hint: Draw stack frames!)

```c
void swap1( int *a, int *b )
{
    int tmp;
    *a = *a - param;
    *a = *b;
    *b = tmp;
}
```