#1. The _________________ SPARC instruction causes the current register set window to slide down 16 registers within the Register Bank. When this instruction completes execution, the old register set registers %_____ — %_____ are now mapped to the new register set registers %______ — %______. The _________________ SPARC instruction saves the current value of %pc into %o7. The _________________ SPARC instruction adds ____ to the value in %i7 and sets %pc with the result.

Using the Rt-Lt Rule, define a variable named foo that is an array of 17 elements where each element is a pointer to a function that takes a pointer to a char as its single argument and returns a pointer to a float.

#2. a) Convert \(163.625_{10}\) to binary fixed-point and single precision IEEE floating-point representation (expressed in hexadecimal).

binary fixed-point __________________________________ x 2^0

IEEE floating-point ________________________________________ (hexadecimal)

b) Convert \(0xC2F8000\) (single precision IEEE floating-point representation) to fixed-point decimal.

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

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

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>
<tr>
<td>Where c is pointing</td>
<td>____________</td>
</tr>
</tbody>
</table>

If the function above is called 5 times, indicate how many times will `d` be initialized to 42?   _________

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

```c
int mystery( int param ) {
    int local = 13;

    if ( local > param )
    {
        local = local - param;
        printf( "%d\n", local ); /* Output the value of local followed by a newline */
        param = mystery( param + 2 ) + local;
        printf( "%d\n", param ); /* Output the value of param followed by a newline */
    } else {
        printf( "Whoa\n" );
    }

    return local;
}
```

Put answer here