#1.

________ subroutine is a subroutine that is expanded inline or is defined as a macro.

________ subroutine is a subroutine that does not call other subroutines.

________ subroutine is a subroutine that supports recursion.

________ subroutine is a subroutine that generally results in an increased generated code size.

________ subroutine is a subroutine that does not guarantee its args will be evaluated only once.

To access local variables stored on the Runtime Stack in the SPARC architecture, you use a ________________ offset relative to register __________.

In gdb, how do you set a breakpoint at a function named fubar?

______________________________

If we have a retl instruction, the next instruction should be ________________.

#2. a) Convert 107.875_{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 0xC2DDC000 (single precision IEEE floating-point representation) to fixed-point decimal.

fixed-point decimal ____________________________ (decimal / no exponential notation)
#3. Name 2 specific ways the SPARC architecture attempts to reduce the overhead of making a function call.

1) 

2) 

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

```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 );
        printf( "%d\n", param ); /* Output the value of param followed by a newline */
    } else {
        printf( "Stop\n" );
    }

    return local;
}
```