1) Given the following program, reorder the output so that the address values that are printed are sorted from smallest to largest if compiled and run on a Sun SPARC architecture. These lines print out the hex address of the different parts of the program (not the values assigned) with the printf() format specifier %p (pointer). Basically, where do the different parts of a C program live in the run time environment?

```c
#include <stdio.h>
#include <stdlib.h>

void foo( int a )
{
    int b = 42;
    int c = 420;

    /* 1 */ (void) printf( "1:a --> %p\n", &a );
    /* 2 */ (void) printf( "2:b --> %p\n", &b );
    /* 3 */ (void) printf( "3:c --> %p\n", &c );
    /* 4 */ (void) printf( "4:foo() --> %p\n", foo );
}

int
main( int argc, char *argv[] )
{
    static int d;
    int e;
    static int f = 37;
    foo(e);

    /* 5 */ (void) printf( "5:d --> %p\n", &d );
    /* 6 */ (void) printf( "6:e --> %p\n", &e );
    /* 7 */ (void) printf( "7:malloc --> %p\n", malloc(f) );
    /* 8 */ (void) printf( "8:argv --> %p\n", &argv );
    /* 9 */ (void) printf( "9:f --> %p\n", &f );
    /*10 */ (void) printf( "10:argc --> %p\n", &argc );

    return 0;
}
```

This line number would print the smallest value/address

This line number would print the largest value/address

(over)
2) What gets printed if the following function is invoked as `recurse( 7, 4 )`? Hint: Draw stack frames.

```c
int recurse( int a, int b ) {
    int local = a + b;
    int result;

    if ( local >= 8 )
        result = recurse( a - 1, b ) + local;
    else
        result = local;

    printf( "%d\n", result );
    return result;
}
```

3) Order the following storage hierarchy elements/types from fastest to slowest

(A) Registers  (D) L1 cache
(B) L2 cache    (E) RAM (Main Memory)
(C) Tape        (F) Hard Disk

4) Which part of the entire compilation sequence clear through to program execution is responsible for:

(A) expanding # directives
(B) ensuring the bss segment is set up and zero-filled
(C) reporting syntax errors
(D) reporting multiply-defined symbols
(E) creating an executable image from multiple object files
(F) getting the executable image from disk into memory
(G) reporting a relocation truncated to fit: R_SPARC_13 data error
(H) translating assembly source code into object target code
(I) translating C source code into assembly target code
(J) resolving undefined external symbols with defined global symbols across modules
(K) having the operating system report a segmentation fault (cored dumped) message

5) Specify the scope/visibility of each of the following:

(A) global scope
(B) file scope
(C) local/function scope

What question would you like to see on the Final Exam? (1 pt)