#1 Which part of the entire compilation sequence clear through to program execution is responsible for:
- getting the executable image from disk into memory
- translating C source code into assembly target code
- ensuring the bss segment is set up and zero-filled
- resolving undefined external references with defined global references across modules
- creating an executable image from multiple object files
- translating assembly source code into object target code

How many bits are needed in a SPARC Format 3 instruction to a specific register? ______

When a program performs I/O, the system saves all the current information about the running program and schedules another program to use the cpu while waiting for that I/O to complete. This process is known as a ____________________________ .

Which type of linking ensures all the library code used in a program is fully self-contained in the resulting executable? _____________

Order the following storage hierarchy elements/types from fastest to slowest
A) L1 cache  C) Registers  E) Tape
B) Hard disk  D) L2 cache  F) RAM (Main Memory)

____ ______ ______ ______ ______ ______  (Fastest)

____ ______ ______ ______ ______ ______  (Slowest)

#2. What gets printed if the following function is invoked as `recurse( 5, 10 )`? Hint: Draw stack frames.

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

    if ( b > 7 )
        result = local + recurse( a, b - 1 );
    else
        result = local;

    printf( "%d\n", result );

    return result;
}
```

(over)
#3. 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 = 42;
    static int b;
    
    /* 1 */ (void) printf( "1:a --> %p\n", &a );
    /* 2 */ (void) printf( "2:b --> %p\n", &b );
}

int main( int argc, char *argv[] ) {
    static int c = 42;
    int d = 420;
    int e;
    
    foo();
    
    /* 3 */ (void) printf( "3:e --> %p\n", &e );
    /* 4 */ (void) printf( "4:c --> %p\n", &c );
    /* 5 */ (void) printf( "5:d --> %p\n", &d );
    /* 6 */ (void) printf( "6:argc --> %p\n", &argc );
    /* 7 */ (void) printf( "7:malloc --> %p\n", malloc(d) );
    /* 8 */ (void) printf( "8:argv --> %p\n", &argv );
    /* 9 */ (void) printf( "9:foo() --> %p\n", foo );
    
    return 0;
}
```

This line number would print the smallest value/address

This line number would print the largest value/address

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